AN EMPIRICAL METHODOLOGY FOR ENGINEERING HUMAN SYSTEMS INTEGRATION

DISSERTATION

Nicholas S. Hardman, Major, USAF

AFIT/DS/ENV/09-D01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

AFIT/DS/ENV/09-D01

AN EMPIRICAL METHODOLOGY FOR ENGINEERING HUMAN SYSTEMS INTEGRATION

DISSERTATION

Presented to the Faculty

Department of Systems and Engineering Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Nicholas S. Hardman, BSEE, MSEE, MSAS

Major, USAF

December 2009

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/DS/ENV/09-D01

AN EMPIRICAL METHODOLOGY FOR ENGINEERING HUMAN SYSTEMS INTEGRATION

DISSERTATION

Nicholas S. Hardman, BSEE, MSEE, MSAS
Major, USAF

Approved:

John M. Colombi, PhD (Chairman)	10 Nov 09 Date
David R. Jacques, PhD (Member)	10 NOV 09 Date
Raymond R. Hill, Jr., PhD (Member)	10 Nov 09 Date
Janet E. Miller, PhD (Member)	10 Nov 09 Date

Accepted:

M. U. Thomas, PhD Dean, Graduate School of Engineering and Management	7 Dec 09 Date

AFIT/DS/ENV/09-D01

Abstract

The systems engineering technical processes of requirements analysis, functional allocation and design synthesis are not sufficiently supported by methods and tools that quantitatively integrate human considerations into early system design. Because of this, engineers must often rely on qualitative judgments or delay critical decisions until late in the system lifecycle. Studies reveal that this is likely to result in cost, schedule, and performance consequences.

This dissertation presents a methodology to improve the application of systems engineering technical processes for design. This methodology is mathematically rigorous, is grounded in relevant theory, and applies extant human subjects data to critical systems development challenges. The methodology is expressed in four methods that support early systems engineering activities. First, a requirements elicitation method guides the transformation of aircraft mishap data, encoded with the Human Factors Analysis and Classification System (HFACS), to prioritized human systems integration (HSI) domain risk areas for a target system. The second method defines empirical functional allocation between humans and automation. The method is used to analyze the collision avoidance capability for a remotely piloted vehicle (RPV). The application shows the limitations of both humans and automation for collision avoidance. The third method is a design approach for input devices, employing a multi-objective nonlinear optimization algorithm to find the input device controller gains based on the performance of a total system model (including the human operator). It makes use of a simulation-based approach accommodating empirical data for human capabilities and limitations. The final method guides the layout of

information in multi-function displays (MFD). This research proposes a human-computer interaction (HCI) index that allows for a quantitative evaluation of layout effectiveness. This was combined with Markov chains and hybrid seeded genetic algorithms to not only evaluate, but find a mathematically best display layout. Algorithm results were confirmed with human subjects test data for F-15 and A-7 avionics. These methods form a coherent approach to early system development.

Each method is separately discussed and demonstrated using a prototypical system development program – an advanced multi-role RPV. In total, this original and significant work has a broad range of applicability to improve the engineering of human systems integration.

To SSAST; the home team.

Acknowledgments

I would like to express my appreciation to Drs. Colombi, Jacques, Hill and Miller; who showed how to integrate this human into the system that is academia. I know that together we have made some significant advances in systems engineering, and along the way I have grown in understanding regarding your varied individual disciplines. I would also like to thank AFRL for the support of this research.

I am also indebted to my friend for the last decade, and mother of our children, for her patience and corresponding study.

Nicholas Hardman

Table of Contents

List of Figures

Page

List of Tables

List of Abbreviations

ACAS .... Automated Collision Avoidance System
AF.. Air Force (US Air Force)
AFDD................................. Air Force Doctrine Document
AFI .... Air Force Instruction
AFMC................................. Air Force Materiel Command
AFPD................................. Air Force Policy Directive
AFRL .... Air Force Research Laboratory
AFSC.................................. Air Force Safety Center
AMC.................................. Air Mobility Command
ANSI .... American National Standards Institute
ASC .... Aeronautical Systems Center
ATC.................................... Air Traffic Contol
ATM................................... Air Traffic Management
ATS .... Air Traffic Service
AZ .... Azimuth
CAS .... Close air support
CBRN .... Chemical, biological, radiological, and nuclear
CHI .... Computer-human interaction
CI... Configuration item
COA.................................... Certificate of Authorization
CRM................................... Crew (Cockpit) resource management
CSE .... Center for Systems Engineering
DAG .... Defense Acquisition Guidebook
DARPA .... Defense Advance Projects Agency
DAU .... Defense Acquisition University
DoD.................................... Department of Defense
DoDAF .... DoD Architectural Framework
DoDD .... Department of Defense Directive
DoDI .... Department of Defense Instruction
DSA..................................... Detect, see, and avoid
EIA .... Electronic Industries Alliance
EL.. Elevation
FAA..................................... Federal Aviation Administration
FAR..................................... Federal Acquisition Regulation
FCS...................................... Flight control system
FOR .... Field of regard
FOV .... Field of view
FY.. Fiscal year
GAO .... Government Accountability Office
GEIA .... Government Electronics & Information Technology Assoc.
HAZOP.............................. Human Error Hazard and Operability study
HCC .... Human centered-computing

HCI .... Human-computer interaction or Human-computer interface
HE....................................... Human Engineering (MILSPEC name for human factors)
HEIST .... Human Error In Systems Tool
HET.................................... Human Error Taxonomy
HF .... Human factors
HFACS .... Human Factors Analysis and Classification System
HFE .... Human factors engineering
HFI...................................... Human factors integration
HMI .... Human-machine interface; avoids gender issue of MMI
HSE.................................... Human systems engineering
HSI .... Human system integration, also: human system interaction
ICAO .... International Civil Aviation Organization
ID .... Index of difficulty
IEC..................................... International Electrotechnical Commission
IEEE................................... Institute of Electrical and Electronics Engineering
IFR .... Instrument flight rules
IIE .... Institute of Industrial Engineering
IMC.................................... Instrument meteorological conditions
INCOSE............................. International Council on Systems Engineering
ISO .... International Organization for Standardization
ITAA.................................. Information Technology Association of America
IxD or IaD .... Interaction design
JCIDS................................. Joint Capabilities Integration Development System
JCS...................................... Joint Chiefs of Staff
JROC .... Joint Requirements Oversight Council
KSA.................................... Knowledge, skills, and abilities
LADAR .... Laser detection and ranging
LIDAR............................... Light detection and ranging
MFD .... Multi-Function Display
MIL-HDBK....................... Military Handbook
MIL-SPEC .... Military Specification
MIL-STD .... Military Standard
MMI .... Man-machine interface
MNS................................... Mission Needs Statement
MPT .... The HSI domains of manpower, personnel, and training
MQ-X .... Next Generation UAS
NAS.................................... National Airspace System
NASA .... National Aeronautics and Space Administration
NDIA................................. National Defense Industrial Association
NM..................................... Nautical miles
NSS .... National Security Strategy
OASys................................. Obstacle awareness system
ORD .... Operational Requirements Document
ORM.................................. Operational risk management
OSD .... Office of the Secretary of Defense

PM....................................... Program manager
PVI, UVI, OVI.................. Pilot (user, operator) vehicle interface
RCS .... Radar cross section
ROA.................................... Remotely operated aircraft
ROC .... Receiver operator characteristic curve
RPV..................................... Remotely Piloted Vehicle
SAA..................................... Sense and avoid (alternate term DSA)
SE .... Systems engineering, systems engineer
SHERPA .... Systematic Human Error Reduction and Prediction Approach
SIB....................................... Safety investigation board
SMC .... Systems, Man, and Cybernetics Society of IEEE
SPO .... System program office
UAS..................................... Unmanned aerial system
UAV .... Unmanned aerial vehicle
UI .... User interface
UNICORN .... Universal collision obviation and reduce near-miss system
USAF .... US Air Force
VFR..................................... Visual flight rules
VMC.................................... Visual meteorological conditions

List of Symbols

A_e	Effective antenna area
$\boldsymbol{A}(G)$	Adjacency matrix of a graph
b	Flight path
c	Speed of light
D	Detector aperture diameter
$d^{-}(v)$	Indegree of a vertex v
$d^{+}(v)$	Outdegree of a vertex v
$dia^{-}_{w}(G)$	The weighted in-diameter of an HCI-defined graph
$d_w(v_0,v_k)$	Weighted distance from v_0 to v_k on an HCI-defined graph
$D_{w,p}(G)$	HCI-Index
$E(G)$	The edge set of a graph
$\mathrm{ecc}_w(\nu_a)$	The weighted eccentricity of a vertex v_a in a graph
$e_{a,b}$	Edges (lines) of a graph
f	Frequency
G	A graph
G	Antenna gain
g	Force of gravity
G	Forward transfer function
G_n	Central nervous system gain
H	Feedback transfer function
H_s	Information content of task
K	Control system gain
K_s	Human operator strategy transfer function
N	Information content
n	Load factor
n	The cardinality (size) of the vertex set
P_{avg}	Average power
P_{XMTR}	Power in the transmission path of the laser
R	Detection range
R	Range to target
r	Turn radius
r_f	Miss distance (radius)
s	Trajectory
S_{min}	Minimum detectable signal energy
T	Closed-loop transfer function
t_{c-}	Necessary warning time
T_d	Reaction time delay
t_f	Final time, for collision
T_m	Movement time delay
t_{ot}	Time on target, dwell time, or integration time
v	Velocity
v_x	A vertex (or node) of a graph

$V(G)$ .... The vertex set of a graph
v_c .... Closure velocity
$W(G)$ .... Wiener Index
W_R .... Linear resolution
α .... Probability of a Type I error
α_M .... Movement informatic parameter (Hz)
β .... Choice reaction time baud rate
β .... Probability of a Type II error
$\Delta^{-}(G)$ .... Maximum indegree of a Graph G
$\delta^{-}(G)$ .... Minimum indegree of a Graph G
$\Delta^{+(}G)$ .... Maximum outdegree of a Graph G
$\delta^{+}(G)$ .... Minimum outdegree of a Graph G
η_{ATM} .... Transmissivity, atmospheric
η_{RCVR} .... Transmissivity, receiver path
η_{XMTR} .... Transmissivity, transmission path
Θ .... Elevation angle
Θ_R .... Angular resolution
λ .... Wavelength
$\varrho_{a,b}$ .... Probability of transitioning an edge in a graph
ρ_T .... Target reflectivity
$\boldsymbol{P}$ .... Affinity matrix of a graph
σ .... Target radar cross section
ϕ .... Bank angle
Ψ .... Azimuth angle
ω .... Turn rate

AN EMPIRICAL METHODOLOGY FOR
ENGINEERING HUMAN SYSTEMS INTEGRATION

I. Introduction

The ideal engineer is a composite; he is not a scientist, he is not a mathematician, he is not a sociologist or a writer; but he may use the knowledge and techniques of any or all of these disciplines in solving engineering problems. –N. W. Doughert

Overview

Systems engineers do not have sufficient means to quantitatively integrate human considerations into system development. Though human systems integration (HSI) is a growing segment of systems engineering (SE) literature, studies reveal that many projects still fall short of the system effectiveness that is achievable if human components are fully integrated in the systems engineering processes (Bainbridge, 2004; Bausman, 2008; Bias & Mayhew, 2005; Booher, 2003; Dekker, 2004; Dray, 1995; GAO, 2005; Harris & Muir, 2005; Malone & Carson, 2003; Mayhew, 1999; Norman, 2007; USAF HSI Office, 2008).

The foundational research for this work is a study of current issues in systems development, a review of U.S. Department of Defense (DoD) acquisition policy, and an analysis of current HSI literature. It forms the motivation for the proposed methodology to engineer human systems integration in system development.

The remainder of this chapter lays out the problem statement, originality and significance of this work, previous publications by the author, and the scope and assumptions. This is followed by background information.

Problem Statement

After a literature review of the challenges to successful system development, it is hypothesized that the process can be improved with a methodology that considers the human component more completely and quantitatively. Therefore, the following research questions were pursued.

1. What must new methods and tools deliver if they are to help address the issues that challenge system development?
2. Given that the human component is central to many of the issues, how must human considerations be better incorporated into SE technical processes?

Originality and Significance

This research develops an original methodology grounded in the established mathematics of graph theory, statistics, optimization, and control theory and draws from empirical research in information theory, physiology, and applied psychology. Since the problems of HSI are complex and broad, it is logical to seek to address them with this multidisciplinary synergistic method. This approach is a significant contribution to systems engineering as it overcomes the weaknesses of current methods which are generally qualitative, do not provide traceability, and do not take full advantage of the depth of data on human capabilities and limitations.

Contributions and Publications

As part of this research, I, in co-authorship with the members of my research committee, have written ten papers for peer-reviewed venues. Seven papers were submitted for conference or periodical publication and three were submitted for journals. The first conference paper was published by the Institute of Electrical and Electronics Engineers (IEEE) as part of the 2008 International Conference on Distributed Human-Machine

Interfaces. That paper studies an original approach to designing display layouts using graph-theoretic modeling and quantitative task analysis (Hardman et al., 2008a). This work is contained in Appendix D in support of Method 4 discussed in this dissertation. The second paper was published in the April 2008 edition of INSIGHT, a quarterly publication of the International Council on Systems Engineering (INCOSE). It discusses how to apply Human-Computer Interaction (HCI) tools and principles to perform systems engineering technical processes (Hardman et al., 2008b). This is contained in Appendix E as one of the other topics of interest. The third paper was also published in INSIGHT and it clarifies terminology and conceptual approaches for complex systems engineering (Narkevicius, Winters, & Hardman, 2008). This is contained in Appendix E as one of the other topics of interest. The fourth paper was published by the Institute of Industrial Engineers (IIE) as part of the 2008 International Engineering Research Conference. This was an invited paper that discusses how to model HSI issues within the DoD Architecture Framework (DoDAF) (Hardman et al., 2008). This is contained in Appendix E as one of the other topics of interest. The fifth paper was published in the Conference on Systems Engineering Research (CSER), 2009. This paper discusses current SE challenges and the connection to HSI (Hardman et al., 2009b). This is contained in Appendix E as one of the other topics of interest. The sixth paper was published in the Proceedings of the INCOSE Annual International Symposium 2009. This paper discusses the challenges of integrating the human component in the SE Technical Processes (Hardman et al., 2009c). This is contained in Appendix E as one of the other topics of interest. The seventh paper was published by IEEE for the 2009 Systems, Man, and Cybernetics (SMC) International Conference. That paper uses graph theoretic modeling in a hybrid genetic algorithm to optimize display layouts

(Hardman et al., 2009a). This is contained in Appendix D in support of Method 4 discussed in the paper.

Of the three journal papers, the first was for the IEEE SMC Journal, Part A. That paper presents the complete design approach for display layouts and performs a validation using real human subjects data (Hardman et al., submitted for publication). This is Method 4 as discussed in Chapter 7. The second journal paper is for the American Institute of Aeronautics and Astronautics (AIAA) Journal of Aerospace Computing, Information, and Communication. That paper presents a method for requirements elicitation through mishap analysis and demonstrates its application (Hardman et al., submitted for publication). This is Method 1 as discussed in Chapter 4. The third journal paper was written for the International Journal of Applied Aviation Studies. That paper presents the method for empirical function allocation and its application (Hardman et al., accepted for publication). This is Method 2 as discussed in Chapter 5.

In addition to the above mentioned publications, the method and related tool described in Chapter 6 has been accepted by Mathworks© as a MATLAB© User Community Application.

Scope and Limitations

This research focuses on improving the DoD systems engineering technical processes for design based on USAF and aviation-centric data; however, the DoD SE community has co-evolved their processes with the global systems engineering community. Therefore, the reader should see that the methodology readily generalizes to other communities if it is used

in the context of their process activity roadmaps such as those described in the INCOSE SE Handbook, IEEE 1220, or ISO 15288 (IEEE, 2008; INCOSE, 2007; ISO/IEC, 2007).

Background

Before engaging in a discussion of how to improve systems engineering technical processes, some background information is needed.

Systems Engineering

As defined by INCOSE, systems engineering is "an interdisciplinary approach and means to enable the realization of successful systems" (2007). Systems are defined by the American National Standards Institute (ANSI) and the Electronic Industries Alliance (EIA) Standard 632 as "an integrated composite of people, product, and processes that provide a capability to satisfy a stated need or objective" (2003). The Defense Acquisition University (DAU) uses a similar definition and describes SE as being "the integrating mechanism across the technical efforts related to the…lifecycle processes" (2006). Thus, system engineers address how the parts fit within the whole and manage the impact of complexity and change in system development.

Systems engineering is often illustrated using a 'V', or overlapping sequence of 'V's. The version of the "SE Vee", shown in Figure 1, illustrates the relationship of key systems engineering activities. As Buede asserts, this interaction is conducted, iteratively and at all levels of component development and integration (2000). As mandated by DoD Instruction (DoDI) 5000.02, all DoD acquisition programs are expected to use sound systems engineering processes. This must be demonstrated in a systems engineering plan (SEP) that is reviewed at each milestone decision (DoD, 2008).

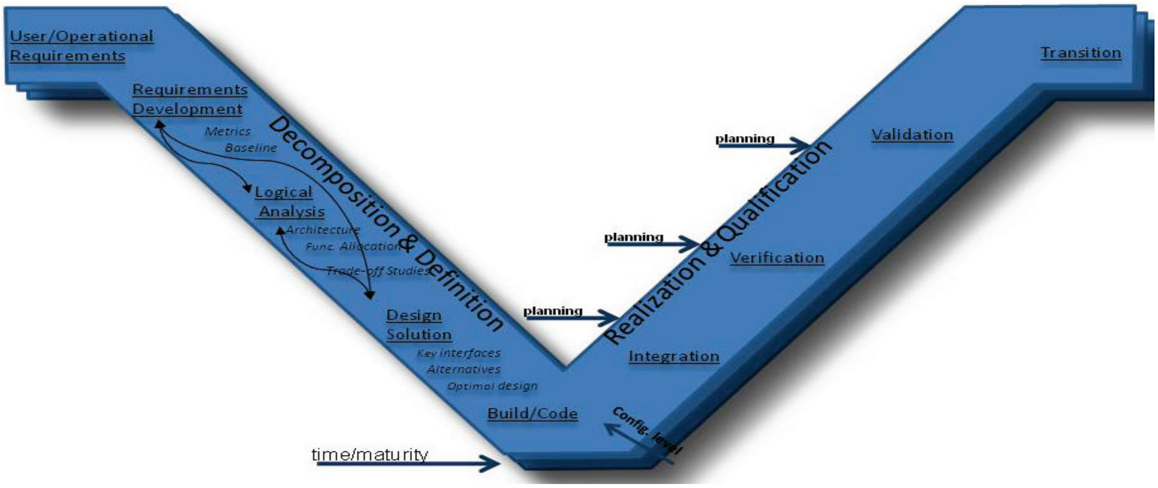


Figure 1. SE Processes on the SE Vee Model

System Lifecycle

Systems engineering encompasses the entire system lifecycle, from concept to disposal. The ISO 15288 defines the six stages of a system lifecycle as: concept, development, production, utilization, support, and retirement (IEEE, 2008). For the DoD, the revised acquisition lifecycle has been defined in DoD Instruction (DoDI) 5000.02 as shown in Figure 2. This guidance reflects the increased focus on work that occurs prior to official program initiation at Milestone B (DoD, 2008). .

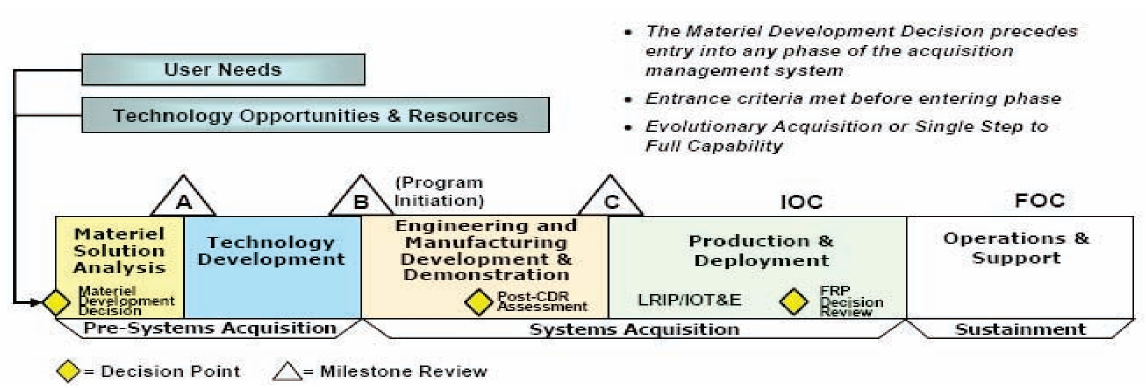


Figure 2. New DoD Acquisition Lifecycle
(DoD, 2008)

Human Systems Integration

Human systems integration (HSI), as defined by INCOSE, is "the interdisciplinary technical and management processes for integrating human considerations within and across all system elements; an essential enabler to systems engineering practice" (2007). HSI is concerned with providing methods and tools that support the SE community by ensuring humans are considered throughout the systems development process in a logical and effective way (Pew & Mavor, 2007). HSI is often expressed in terms of its functional domains. Figure 3 presents the nine domains of HSI in the official nomenclature of the US Air Force (AIRPRINT, 2005). Systems engineers do not replace the disparate organizations of each domain, but they must effectively interact with them.

Document Overview

The next chapter presents a review and analysis of background literature. Chapter 3 presents the methodology for engineering HSI in the SE technical processes. Chapters 4

Figure 3. HSI Domains
(AIRPRINT, 2005)

through 7 are semi-independent sections that each describe and demonstrate one of four methods developed by this research. Each method applies the methodology to particular activities in the SE technical processes. Chapter 8 contains conclusions and recommendations and proposes future research. These chapters are supplemented by five appendices; each appendix contains original products of the research which support a particular section of the dissertation. They were not included in the main body so as not to interrupt the flow and readability of the document.

II. Background Literature Review

Life was simple before World War II. After that, we had systems. –G. Hopper

Overview

This chapter captures the review and analyses of applicable literature on systems development and human systems integration. It also contains a summary of relevant DoD guidance and recognized international standards for systems engineering. The chapter reveals what significant issues continue to challenge system development, and the involvement of HSI in those issues. This review forms the foundation and the impetus for the new methodology presented in Chapter 3.

Systems Engineering Processes

Systems engineers develop systems using prescribed processes. These are interacting activities which transform inputs into outputs (ISO, 2005). Various communities have established standards to formalize SE processes. "Standards are meant to provide an organization with a set of processes that, if done by qualified persons using appropriate tools and methods, will provide a capability to do effective and efficient engineering of systems." (DAU, 2006). Standards delineate *what* needs to be done, but they generally do not dictate *how* to do it. The INCOSE SE Handbook expands on the processes and process groups defined by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) in ISO 15288 (INCOSE, 2007). As it states in the first chapter, this standard is intended to be supported by methods and tools developed for particular organizations (ISO, 2008).

Figure 4 shows the evolution of systems engineering standards beginning with the influence of the US military (Martin, 1998). As the DoD acquisition community was reformed and military standards (MIL-STDs) were eliminated, the Government Electronics and Information Technology Association (GEIA), created EIA-632: *Processes for Engineering a System* from MIL-STD-499. GEIA has now merged with the Information Technology Association of America (ITAA) and the newest revision of the standard has been adopted by ANSI. This standard is a high-level look at the processes that should be standardized industry-wide. The ANSI concept for the iterative relationship between formal processes in a system lifecycle is portrayed in Figure 5 (ANSI/GEIA, 2003).

The International Electrotechnical Commission (IEC) and the International Organization for Standardization (ISO) have produced ISO/IEC 15288: *Systems Engineering--System Life Cycle Processes.* They have also produced ISO 19760: *A Guide for the Application of ISO 15288* (IEEE, 2008). Organizations that represent the practitioners have largely accepted these efforts of standardization. The Institute of Electrical and Electronics

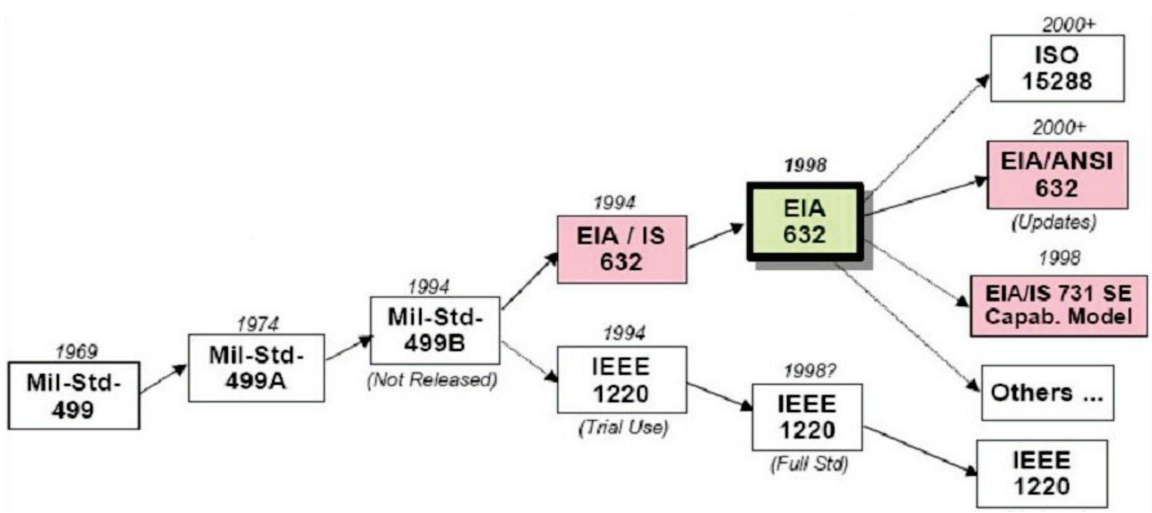


Figure 4. Evolution of SE Standards
(Martin, 1998)

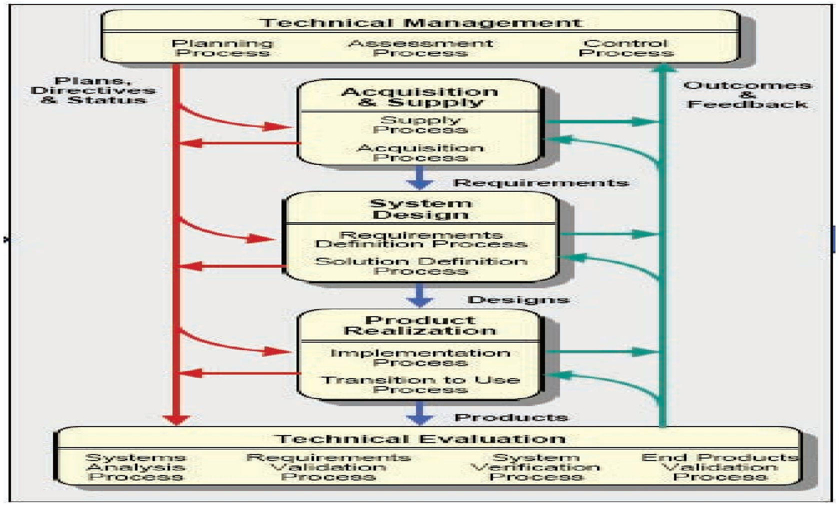


Figure 5. The Iteration of SE Processes, ANSI
(ANSI/GEIA, 2003)

Engineers (IEEE), for example, has adopted the ISO standard as their own IEEE 15288. In addition, they have created IEEE 1220 to standardize the application of systems engineering processes. Recursively, ISO has now accepted the IEEE 1220 as their standard (ISO/IEC, 2007). The INCOSE Handbook also adheres to the technical processes codified in the ISO/IEC 15288 (INCOSE, 2007). These organizations are all active in developing and ratifying global SE standards. They recommend processes that guide systems engineering throughout the system lifecycle and among all specialties and stakeholders. ISO has just updated and released ISO 15288:2008. They are also working towards a more formal interface management process that is fully integrated with configuration management (IEEE, 2008).

It appears that the DoD has been slow to adjust from being a standards maker to a standards conformer; however, efforts are under way to align with the international community. There is active work to update AFI 63-1201: *Life Cycle Systems Engineering* and other DoD SE documents to be consistent with ISO 15288 (Bausman, 2008).

Though the DoD's influence on systems engineering standards may have faded, the requirement for systems engineering in the DoD has not diminished. The DoD Directive (DoDD) 5000.01 mandates the application of a systems engineering approach that optimizes performance and minimizes costs (DoD, 2003a). DoDI 5000.02, paragraph 3.9.2.2 states that "The effective sustainment of weapon systems begins with the design and development of reliable and maintainable systems through the continuous application of a robust systems engineering methodology." (DoD, 2008).

In the 1990s the DoD sought to find better common representations to communicate structures of systems and enterprises. They developed and ratified the Department of Defense Architecture Framework (DoDAF) as a modeling aid to support complex system design. The current DoDAF, Version 2.0, was mandated for use in May of 2009. Architecture is, "the structure of components, their relationships, and the principles and guidelines that govern their design and evolution over time" (DoDAF Working Group, 2009a). For architectures using DoDAF, this structure of components is captured in the underlying logical data model called the DoDAF Meta Model (DoDAF Working Group, 2009b). The HSI domains can be incorporated into the DoDAF. This was explored in (Hardman et al., 2008), which is contained in Appendix E3.

The US Air Force has been a leading voice for the application of systems engineering processes in the DoD. Figure 6 shows the relationship between DoD, Air Force, and command level policies (Jaggers, 2005). The Air Force Policy Directive (APPD) 63-1: *Capability-Based Acquisition Systems* mandates that the Air Force use sound systems engineering processes system acquisition. This is implemented by Air Force Instruction (AFI) 63-1201: *Life Cycle Systems Engineering*, which was updated in 2007 to reflect a more holistic application

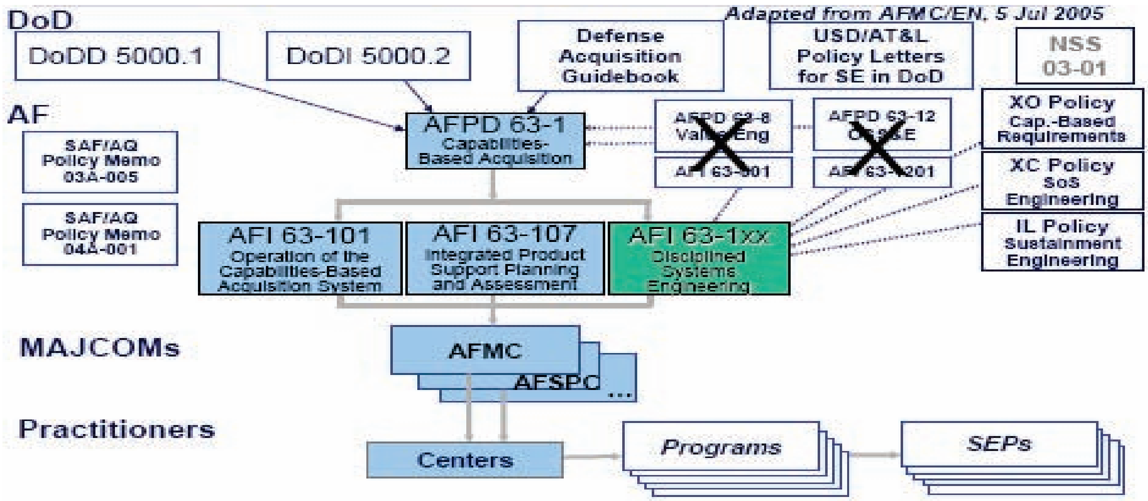


Figure 6. USAF SE Policy Development
(Jaggers, 2005)

of SE processes. The AFI states the senior leadership's commitment to improve the "credibility and rigor of the processes by which (systems) are developed, produced, integrated, tested, fielded, operated, maintained, sustained, and supported" (USAF, 2007). AFMCI 63-1201 is the Air Force Materiel Command's instruction for implementing the instruction of AFI 63-1201. It provides the implementation guidance and standards for system development in the Air Force. In it we find the recommended SE processes for use in Air Force acquisition (AFMC, 2007).

The SE processes defined in ISO/IEC 15288, and expanded upon in the INCOSE SE Handbook, are grouped into four process groups. One of these, the Technical Processes group, involves making technical decisions to optimize the benefits and reduce the risks during system development. These processes consist of: Stakeholder Requirements Definition, Requirements Analysis, Architectural Design, Implementation, Integration, Verification, Transition, Validation, Operation, Maintenance, and Disposal (INCOSE, 2007). DAU provides supplementary information regarding the SE processes identified in DoD

policy. In their Defense Acquisition Guidebook (DAG), they list 16 systems engineering processes divided into Technical Processes and Technical Management Processes. The technical management processes provide control and planning to the entire design effort (DAU, 2006). The ISO and DAU process sets are comparable, but DAU has not yet changed the process names to be consistent with the ISO standard. DAU further divides the technical processes into the Technical Processes for system design (the top-down design, or left side of the Vee) and the Technical Processes for product realization (the bottom-up realization, or right side of the Vee). The Technical Processes for system design consist of Requirements Development, Logical Analysis, and Design Solution. The Technical Processes for product realization consist of Implementation, Integration, Verification, Validation, and Transition (DAU, 2008). This division is consistent with the EIA-632 standard (ANSI/GEIA, 2003).

A visual representation of the relationships of these processes is given in Figure 7 which is the revised SE engine long used by the DAU (CSE, 2008b). It illustrates the iterative relationship among SE technical processes. This iterative flow is intended to be applied at

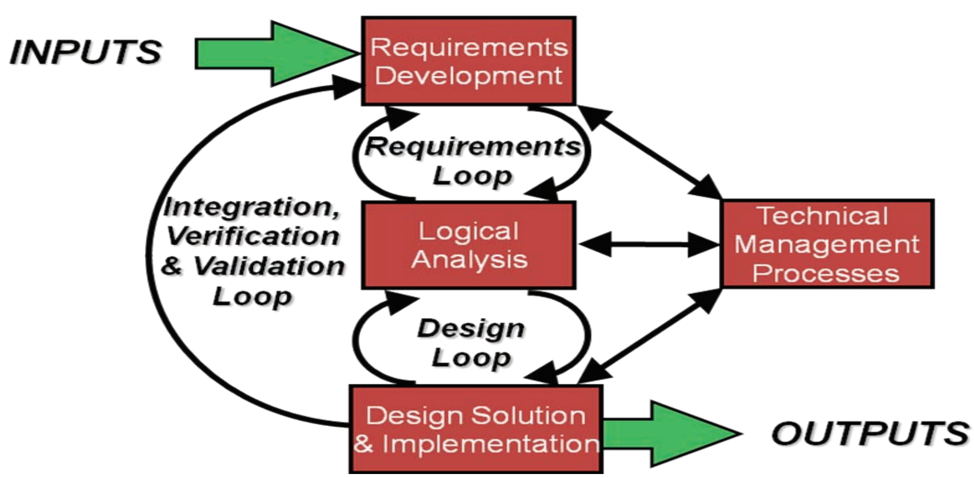


Figure 7. The Iteration of SE Processes, DoD
(CSE, 2008)

all levels of the system development so that the systems engineer can define the boundary of the problem and the top-level requirements and then decompose to sufficient detail for defining feasible solutions. If the Technical Processes for system design are executed with tools and methods that facilitate quantitative analysis, then the Technical Processes for product realization can be performed quantitatively as well. Thus, the focus of this research is on the Technical Processes for system design. The following paragraphs summarize the analysis performed in each process.

Requirements Development

The purpose of the requirements development process is to define the boundary of the problem and the top-level requirements to be satisfied. During this process, the projected mission, context, and technology readiness is evaluated. Stakeholder inputs and legacy systems are used to define the needs and priorities of the new system. These are refined into technical requirements. The constraints on system solutions are also identified (DAU, 2006).

Logical Analysis

The purpose of the logical analysis process is to improve understanding of the technical requirements and their inter-relationships. In this process, top-level requirements and constraints are decomposed and functions are allocated to system components. This creates derived technical requirements and necessary component interfaces. The process enables the completion of system development in a logical manner (DAU, 2006).

Design Solution

The purpose of the design solution process is to translate the outputs of the requirements development and logical analysis processes into feasible alternative solutions.

Then, final design selections are made. This results in a physical design of all system components capable of performing the required functions within the identified constraints. These design decisions must be objective and traceable (DAU, 2006).

Systems Development Challenges

Systems engineering has been generally accepted as a necessary part of acquisition, but studies have exposed development challenges that SE appears to inadequately address (Bias & Mayhew, 2005; Booher, 2003; CSE, 2008a; Malone & Carson, 2003; Pew & Mavor, 2007). Sage and Rouse, in their comprehensive *Handbook of Systems Engineering and Management*, conclude that SE processes are fundamentally sound, but the application of SE processes manifests many common problems. Their handbook contains a list of what they call the "most deadly systems engineering transgressions". Of the most critical, they list: a failure to develop and apply the appropriate methods to support the SE processes, a failure to design the system with consideration for the "cognitive style and behavioral constraints" that affect the users, and a failure to design the system for effective user interaction (1999). Designers fail to design the system for effective user interaction when they forget that users will not see the system as they do. Users will only see what is visible or represented in the user interfaces (Pew & Mavor, 2007).

Recent studies within the DoD have come to similar conclusions regarding deficiencies of systems engineering practice in military acquisition programs (CSE, 2008b; Saunders, 2005; USAF Studies Board, 2008). The consequences of these deficiencies in process execution have been costly and time consuming. A 2005 U.S. Government Accountability Office (GAO) report found that major weapon systems programs experience early cost

increases by an average of 42% over original estimates and schedule slips by an average of nearly 20%. Of the identified overrun causes, the GAO analysts determined that most were the result of problems that could have been discovered early in the design process (2005). Recent DoD assessments indicate that the reason these problems are not being discovered early is that insufficient SE is applied early, requirements are not well managed, and SE tools are inadequate (CSE, 2008a).

The committees reviewing new standards for the ISO have also stated that there is a need for improved tools for requirements measurement and interface management. They desire to improve ISO standards as such tools are developed (Bausman, 2008). A recent interview with engineers in the defense industry reveals that they also desire better technical tools. The engineers were specifically emphatic when discussing HSI requirements. They stated that, without the ability to better capture and express these requirements quantitatively, they will never gain full consideration in program management tradeoff analysis; objective performance measures always dominate when contracts are at stake (Graeber & Snow, 2008). Program managers continue to need actionable data earlier in the acquisition process. A U.S. Air Force HSI Office study has determined that the continued compression of acquisition timelines means that irreversible design decisions will increasingly occur in the earliest stages of development (USAF HSI Office, 2008).

The National Defense Industrial Association (NDIA) performed a study to identify areas of DoD acquisition requiring improvement. They concluded that insufficient SE is applied early in the program lifecycle; thus hindering initial requirements and architecture generation. They also concluded that the tools and methods in use are inadequate to execute the SE processes (2003).

In summary, there has been a plethora of studies investigating the causal factors of programmatic failures. They form a general consensus that the challenges lie in the application of SE processes. Specifically, most identify a combination of the following:

1. A need for sound systems engineering *earlier* in system development
2. A need for more *quantitative* methods and tools to support the application of systems engineering processes
3. A need for more effective management and design of *interfaces*

Human Systems Integration

Issues related to human components of systems are woven in the identified challenges. DoD cost studies have established the significance of HSI in such issues as 40 to 60% of the total system's lifecycle cost (INCOSE, 2007). This reality has led to a new DoDI 5000.02-R Sec 4.3.8 which requires comprehensive management strategies for HSI to assure effective human performance, reduce manpower, personnel, and training (MPT) requirements, and comply with all of the constraints for human operation (DoD, 2008). The DoDD 5000.1 directs the program manager to have a comprehensive plan for HSI early in the program lifecycle, a plan that will be reviewed as part of each milestone (DoD, 2003a).

The need for a focus on HSI will only increase as demands on operators are rising and changing in form. Studies by the Air Force HSI Office show that contemporary operators, with near-ubiquitous computer automation and augmentation, are actually experiencing increased workloads (USAF HSI Office, 2008). Though this seems counter-intuitive, it indicates that the increase in complexity, of mission and machine, is growing faster than technological improvements can alleviate. This is being demonstrated operationally by current unmanned aerial system (UAS) employment. The latest UASs are more automated than they have ever been, but the vehicles also have more capabilities, serve in more varied

roles, and execute in larger numbers than ever before (Lindlaw, 2008). This heightens the need for all domains of HSI to be considered in system development, and those responsible for HSI must make useful contributions to the effort. Burns et al. stresses that, to effectively support acquisition, HSI must be able to address tradeoff analyses as part of disciplined system engineering, and it must produce timely analytic data consistent with the level of detail of design choices being made (2005). Air Force studies attest to the proven benefits of addressing HSI challenges early in system development. Potential benefits include: reduced lifecycle cost, reduced time to fielding, shorter training cycles, improved supportability, reduced logistical footprint, and improved safety (USAF HSI Office, 2008).

HSI Domains

The sections that follow define the HSI domains. There is no universal agreement as to the delineation of domains. As can be seen in Table 1, while there is general agreement over the original domains of manpower, personnel, training, and human factors, some communities include additional domains that have not been fully embraced by others.

Though there is no universally accepted set of domains, there is much in common in the understanding of HSI's scope. The definitions that follow are written in terms that enable a system engineer to delineate system requirements by domain and to perform tradeoff analysis between domains. While these are original, they draw heavily from the definitions put forth by the Human Effectiveness Directorate of the Air Force Research Laboratory (AFRL) and INCOSE (AIRPRINT, 2005; INCOSE, 2007).

Table 1. Domains of HSI

HSI Domains	AIRPRINT AFRL/RH	INCOSE	Defense Acquisition Guide	MANPRINT	IEEE SMC	ACM	UK HFI Program
Manpower	X	X	X	X	X	X	X
Personnel	X	X	X	"Personal Capabilities"	X	X	X
Training	X	X	X	X	X	X	X
Human Factors	X	Human Factors Engineering	X	Human Factors Engineering	X	X	Ergonomics
Safety	X	X	"Safety and Occupational Health"	X		X	X
Health	X	"Occupational Health"		X			X
Survivability	X	X	"Personnel Survivability"	X			
Habitability	X	X	X				
Environment	X	X					

Sources: (ACM, 2006; AIRPRINT, 2005; Booher, 2003; DAU, 2006; HFIDTC, 2006; INCOSE, 2007; ISO/IEC, 2007)

Manpower

The manpower domain determines the number and type of personnel required to operate and support a system. Support includes functions such as maintenance, sustainment, and training. Many civilian organizations call this *human resources*. DoD direction on manpower estimates for major defense acquisition programs is extensive. Program managers must coordinate with the manpower community, and the final manpower estimate is reviewed by the Under Secretary of Defense for Personnel and Readiness (DoD, 1999).

Personnel

The personnel domain determines the knowledge, skills, and abilities and the physical, cognitive and sensory capabilities required of the humans in the system. The personnel

community defines these parameters for the system and determines how to best obtain and maintain an adequate pool of qualified persons. The U.S. Army calls it *personal capabilities* and it is related to *human resources* in civilian organizations.

Training

The training domain determines the necessary infrastructure and system components to provide system personnel with the requisite attributes for optimal system performance. This includes individual and unit training programs, training systems, and retraining schedules.

Human Factors

The human factors domain addresses how to incorporate human characteristics and limitations into system design for optimal usability. The issues of this domain are often divided into the following categories:

Cognitive— response times, level of autonomy, cognitive workload limitations

Physical— ergonomic control design, anthropomorphic accommodation, workload

Sensory— perceptual capabilities, such as sight, hearing, or tactile

Team dynamic— communication and delegation, task sharing, management

As computers become more prevalent, systems engineers recognize the immense significance of the interaction of operators and computers. Formal study of this has matured and expanded in perspective over the last three decades. As discussed in Appendix E2, this study is now generally referred to as human-computer interaction (HCI), though there is no standardization on the terms in use. The DoD has listed standard terms and a taxonomy for these terms in the military handbook (MIL-HDBK) -1908B: *DoD Handbook on Definitions of Human Factors Terms* (DoD, 1999) and MIL-STD-1472: *Human Engineering* (DoD, 2003b). The central emphasis of HCI is the design of effective user interfaces (UIs); that is,

the multi-modal exchanges between a human being and hardware. These interfaces facilitate interaction between human cognition and software logic. This is a critical part of human factors. Figure 8 is a conceptual depiction of human factors, HCI, and UI areas of study. The theories and methods of the HCI community can be very useful in environments where the demand for highly effective operator performance is paramount.

Much of U.S. industry calls this *human factors engineering (HFE)* and European and Asian organizations generically refer to it as *ergonomics*. The methods and tools of this domain are the most mature of all the HSI domains.

The DoD has produced extensive guidance on human engineering. This is a term related to the human factors domain, but used in the DoD to describe the application of anthropomorphic and physiological data for system design. The DoD has created MIL-HDBK-759C: *Human Engineering Design Guidelines* (DoD, 2003b). The primary documents for Human Engineering in aerospace vehicles are: MIL-STD 1472 *Human Engineering*, MIL-HDBK-759C *Human Engineering Design Guidelines*, MIL-STD 1787 *Aircraft Display Symbology*, and the Joint Services Specification Guide JSSG 2010 *Crew Systems*. Together, they are the

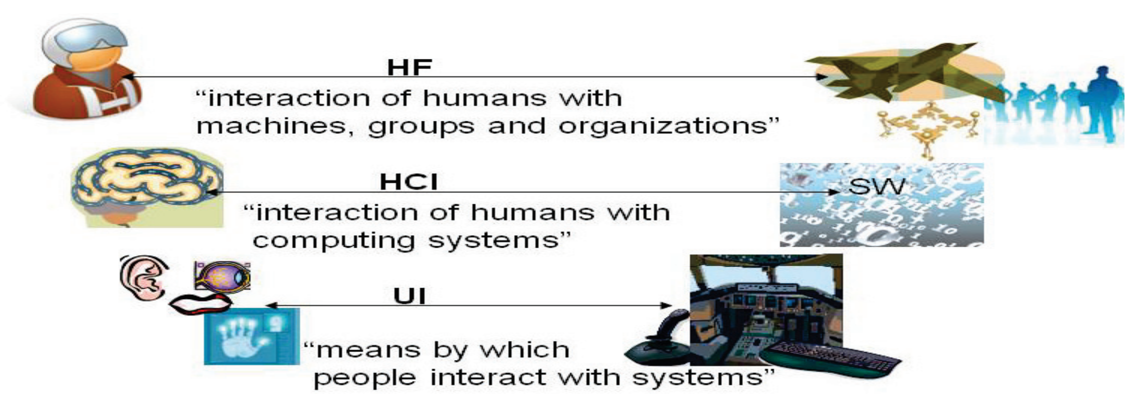


Figure 8. Human Factors Domain
HF—Human Factors, HCI—Human-Computer Interaction, and UI—User Interface

baseline guidance for all human factors design for the US military. For example, MIL-HDBK-759C advises that, in the design of input devices, foot controls should be used only for coarse adjustments and should not be used to adjust a visual display. Hand or arm-operated controls are more desirable for fine adjustment, but if high precision is required, the necessary arm movement should be minimized (DoD, 1998). These requirements are based on valid human subjects research and can be a valuable contribution to SE if they can be better incorporated into methods and tools.

<u>System Safety</u>

The system safety domain evaluates the characteristics and procedures of systems in order to minimize the potential for accidents. Safety studies affect system design by advocating features that eliminate hazards when possible and manage them when they cannot be avoided. Such features include sub-systems for: system status, alert, backup, error recovery, and hazardous environmental exposure.

<u>Survivability</u>

The survivability domain evaluates the characteristics and procedures of systems that can reduce the probability of attack or fratricide, as well as minimizing system damage and injury if attacked.

<u>Health</u>

The health domain evaluates the characteristics and procedures of systems that create significant risks of injury or illness to humans. Sources of health hazards include: noise, temperature, humidity, CBRNE (i.e., chemical, biological, radiological, nuclear, and explosive) substances, physical trauma, and electric shock.

Habitability

The habitability domain evaluates the characteristics and procedures of systems that have a direct impact on personnel effectiveness by maintaining morale, comfort, and quality of life. These characteristics uniquely include: climate control, space layout, and support services.

Environment

The environment domain evaluates the system in the medium for operation. Consideration is made to protect the environment from system manufacturing, operations, sustainment, and disposal activities. In some communities this domain is not considered part of HSI, but rather of systems engineering as a whole.

HSI: Requirement versus Constraint

When incorporating HSI domains into SE technical processes, not all domains should be addressed the same. The human factors, manpower, personnel, and training domains can be effectively managed as design variables that interact with other system requirements. Issues within these domains are often inextricably connected, and system tradeoff analyses must include those inter-relations. For example, as described above, program managers must make early estimates regarding manpower requirements. However, the manpower estimate for a system under development is highly sensitive to decisions made in the personnel, training and human factors domains. This means the manpower estimate must concur with the analysis contained in many tangential documents throughout the system development process.

Conversely, the domains of safety, survivability, health, habitability, and environment form constraints on the system. The analysis of requirements and constraints due to human

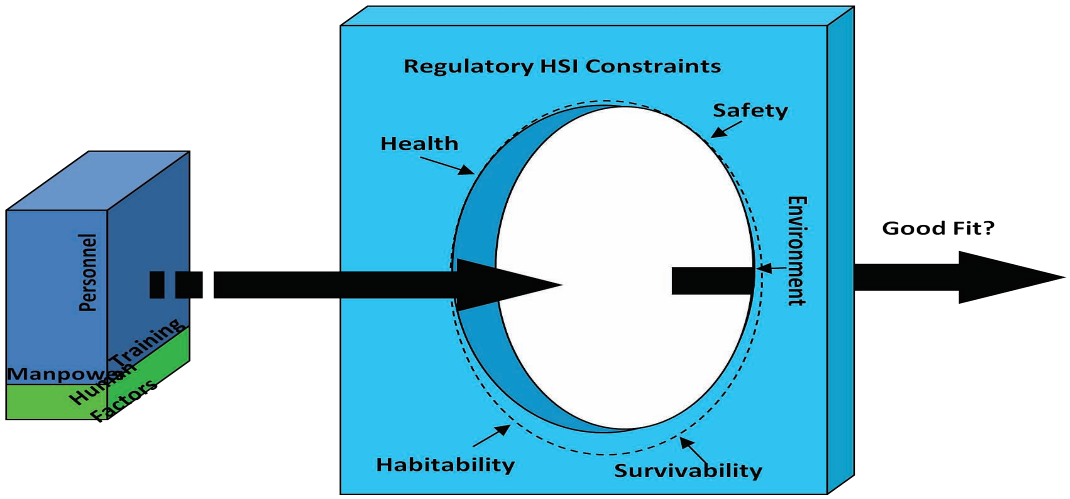


Figure 9. HSI Requirements and Constraints

considerations is a multi-dimensional optimization problem. A useful analogy is to consider these domains in a physical volume sense as portrayed in Figure 9. Analogies allow one to approach new problems in a familiar context. This analogy views the necessary performance of the system as a volume with MPT dimensions. Though the volume is set, the shape is determined by tradeoff decisions. Within the context of the system's intended mission, the constraining domains limit the size and shape of the trade space. Systems engineers must meet the mission requirements, the volume, in such a way that it fits within the contextual constraints, the space. The figure portrays human factors as a fourth component of the volume. Human factors have the effect of either increasing or decreasing the volume. That is, poor human consideration can increase demands in one or more of the volume dimensions; alternatively, improved human factors can be an effective force multiplier by reducing operator workload.

HSI in SE Challenges

The challenges to systems engineering execution are significantly determined by the difficulty of achieving human systems integration. The following sections look at the connection between HSI and the primary challenges to SE process application that were identified previously.

Early HSI

New system development efforts encounter many early HSI-related decisions that commit the design effort to a nearly irreversible course. The Committee on Human Factors of the National Research Council recently undertook a study of the current approaches for HSI in system design. They identified that the SE community needs a better set of methods and tools for early tradeoff analysis (Pew & Mavor, 2007). This was a progressive idea for the human factors community because its traditional focus has been the evaluation of specific components. This expanded view emphasizes an earlier and broader application for methods and tools; ones that could be used to evaluate whole system alternatives and minimize programmatic risk.

Quantitative HSI

Blanchard and Fabrycky say that the improvement path for SE processes requires that they first become more quantitative and then optimized (2006). The HSI-related system requirements are not easily quantified and the tradeoffs are often analytically complex. They are especially difficult to quantify early in the system lifecycle. For example, as explained in the next section, the usability of a system can be standardized and quantified with the use of the proper benchmarks and surveys. This has improved the analysis of these requirements,

but usability evaluations cannot be performed until actual systems, or at least prototypes, have been designed and built.

Interface Management and Design in HSI

An interface is a boundary or point common to two or more entities at which necessary information flow takes place (Booher, 2003). The DoD recognizes that interface management, including the UI, is a key to effective system integration (DAU, 2006). Maier argues that one of the key contributions that SE should make to a program is an attention to the system's interfaces. He offers the following heuristic, "The greatest leverage in system architecting is at the interfaces. The greatest dangers are also at the interfaces." (1999). Maier's heuristic aligns with evidence cited in the text, *Designing for the User Interface* regarding the interfaces of computer and machine (Shneiderman & Plaisant, 2005). Dray identifies a direct connection between lifecycle costs and investment in user interfaces. She cites a company project in which an improved user interface on a large-scale internal application resulted in a 32% overall rate of return due to a 35% reduction in training and a 30% reduction in supervisory time (1995). Savings such as these continue throughout the life of a system and become a significant factor in controlling total ownership cost. Given this criticality, systems engineers must have methods to properly manage UI design.

HCI experts call the measure of a user interface its *usability*. Various sources expand on the concept of usability in different ways, but it is common to define the measure using the following five components (Wickens et al., 2004):

Reliability -- The frequency of errors, the prevention of catastrophic errors, and the ability to recover from errors.

Efficiency -- The level of productivity achieved once learning has occurred.

Learnability -- The amount of training necessary before the user can be productive.

Memorability -- The ability for an infrequent user to maintain proficiency.

Satisfaction -- The subjective experience of the user.

These last three components are unique to the user interface. Because we cannot yet simply program humans, interface learning time and recall ability are essential interface metrics. Standard HCI texts, such as the *Berkshire Encyclopedia of HCI*, address how to measure the five components of usability (Bainbridge, 2004). As discussed further in Hardman, et al., these usability metrics can be standardized and quantified for use in system development (2008b). While usability evaluations make UI analysis quantifiable, designers have a need to predict the potential usability of a configuration even earlier in system development.

Summary

This chapter summarized the current system development challenges and examined how HSI is a pervasive factor in these challenges. To be more effective, systems engineers need better methods and tools to incorporate HSI into the overall technical approach; things that would help them better perform requirements elicitation, function allocation, and design analysis. In Chapter 3, a methodology is presented that can be used early and throughout the SE Technical Processes to achieve that aim.

III. Methodology

In preparing for battle I have always found that plans are useless, but planning is indispensable. –General Dwight D. Eisenhower

Overview

The SE technical processes will be more effective if the challenges identified in the previous chapter are satisfactorily addressed. The processes need supporting methods and tools to provide the means for successful system development. Methods are systematic ways of doing things, and some methods involve tools to perform specific steps of the process. Experts feel that existent methods are too subjective and they do not take full advantage of the depth of data on human capabilities and limitations (Burns et al., 2005).

This research proposes a methodology (i.e., a set of methods and tools with a unifying basis) that will enable sound decision making early in the technical processes. The unifying basis is applying empirical data of human characteristics and limitations to the critical challenges of system development. Theory is essential to science. The scientific method involves creating theories that define phenomena and identify the causal relationships that explain the phenomena. In this way, theories explain what is known and predict what is unknown (Bainbridge, 2004). Work to produce unifying theories in SE has been difficult (von Bertalanffy, 1968). There is a common mantra that, "Systems engineering does not have an *F=ma*" (anon.); that is, the relationships between entities are not so elegantly connected. To date, the community primarily works in principles, heuristics, and other qualitative methods. Though a theoretical foundation of SE has proven elusive, there exist enormous volumes of empirical data regarding all domains of HSI. Blanchard and Fabrycky

say that the improvement path for SE processes requires that they first become more quantitative and then optimized (2006). The engineering application of empirical data for the purpose of integrating humans in systems will help achieve this goal.

Empirical methods are not new to human factors; anthropometric studies have been used to create ergonomic design standards for decades. The archived data from those studies allow developers to make objective decisions early in new system development. The same is possible for other aspects of human consideration in system development. For example, usability analysis cannot be performed until actual systems or prototypes are developed, but empirically-based human data models can make projections of UI usability much earlier. This is akin to the use of fluid dynamics models based on wind tunnel tests to predict the performance of an aircraft body before an actual one is built. Analogous human subjects information could equip designers to effectively address human component issues earlier in system development than is currently possible. To do that, methods using this data must be embedded within the SE technical processes for design.

Requirements Development

During requirements development, key insights can be gained early in the process by an analytic study of legacy system mishaps. Using Method 1, described and demonstrated in Chapter 4, the development team can elicit and prioritize requirements related to the human components of the system. This method may also identify additional human subjects research that will be necessary to support the new system development program.

Logical Analysis

As part of the logical analysis process, a significant amount of effort is required for system decomposition and function allocation. The proper mix and levels of autonomy and manual operation is currently a process with more art than science. Method 2, described in Chapter 5, addresses this activity. It demonstrates that empirical data on human and machine capabilities can be used to quantitatively analyze automation.

Design Solution

During the design solution process, the feasibility of alternatives must be determined and ultimately the final design solution must be picked. For human component issues, this process will often be done iteratively with the function allocation method to determine the optimal mix of user control and the necessary UI requirements. Additionally, user interfaces must be designed for mission context. These are complicated steps that must be done early in system development. This process needs the support of methods and criteria such as those described in Chapters 6 and 7. Method 3 uses models of human performance to improve input device design. Method 4 applies empirical data of human cognition to improve the layout of information in display devices.

Methodology Summary

This empirical methodology for engineering the integration of human components is an original design approach that bridges empirical research with critical design challenges. This resolves the weaknesses of extant methods and enables improved decision making early and throughout the SE technical processes.

IV. Method 1: Requirements Elicitation through Legacy Mishap Analysis[1]

In this chapter, the methodology presented in Chapter 3 is applied to the requirements development process of a prototypical acquisition program. It is shown that the use of the proposed method supports early requirements elicitation.

Introduction

Human error is now deemed the primary risk to flight safety. Studies report that between 60 and 75% of all aircraft accidents involve human-related actions (Dekker, 2006; Eamonn, 2004; Fiorino, 2006; Li, 2006; Shappell & Wiegmann, 2003; Shappell, Detwiler, Holcomb, Hackworth, & Wiegmann, 2007; Stanton et al., 2006; Wiegmann & Shappell, 2001; Wiegmann & Shappell, 2003b). This includes not just the operators in direct aircraft control, but also the humans that, as part of the overall system, perform training, maintenance, and supervision.

Human error has been a highly studied area for decades. At a philosophical level, some deny that true human error really exists, and others believe that it is the root cause of all great tragedies (Dekker, 2004). At an operational level, researchers have greatly improved our understanding of the phenomena. We now have much more sophisticated human error models and error taxonomies regarding the cognitive, perceptual, physiological, and, more recently, organizational aspects. Researchers now advocate that a hazard becomes an accident through the ill-fated alignment of certain latent conditions within the layers of protection in place to prevent such accidents (Reason, 1990). While active errors have

[1] The contents of this chapter have been independently submitted for publication by Hardman, Jacques, Colombi, Hill, and Miller in the AIAA Journal of Aerospace Computing, Information, and Communication..

immediate impact on the system, latent conditions are removed in time and space from the actual event. Figure 10 includes a graphical representation of this accident causation model, commonly called the "Reason Model", after Dr. James Reason, or the "Swiss cheese model". Reason advocates that a focus on these latent conditions holds the most promise for safety improvement (Reason, 1997).

Investigators in all industries have been motivated by this accident causation model to take a deeper look at human error and its causal relationship with accidents. In aviation, this has spawned a theoretically-derived human error framework called the Human Factors Analysis and Classification System (HFACS). Since fiscal year (FY) 2004 the DoD safety community has implemented an HFACS-based taxonomy for use in its aviation accident investigation process. By the end of FY 2010 the DoD plans to release a revised human error taxonomy and implement it in all accident investigations (Musselman, 2009). As Figure 11 shows, the DoD-HFACS identifies four tiers of active failures and latent conditions: Acts, Preconditions, Supervision, and Organization. These are each decomposed into categories and then specific sub-codes (AFSC, 2007). For descriptions of all 147 DoD-HFACS codes, refer to (Hardman & Colombi, 2009).

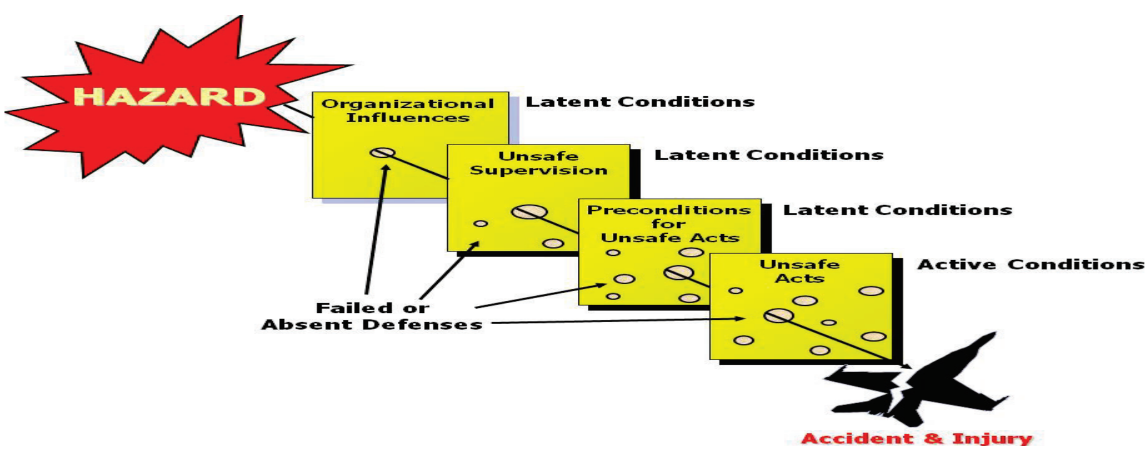


Figure 10. Accident Causation Model
(AFSC, 2007)

HFACS has allowed for a deeper level of accident investigation. Instead of "human error" being seen as the root cause, it is now viewed as the beginning of an investigation into the human-machine interaction breakdown. HFACS allows investigators to characterize a mishap, as in the Reason model, as a function of latent failures. It begins with the active failures and then looks for latent conditions and latent factors in the environment. HFACS has been used to study mishaps across all branches of the military (Thompson, Tvaryanas, & Constable, 2005), in commercial aviation (Li, 2006; Wiegmann & Shappell, 2001), and in multiple recent UAV mishap studies (Thompson et al., 2005; Tvaryanas, 2006a; Williams, 2006). HFACS has also been used to quantify human contributions in maritime shipping accidents (Celik & Cebi, 2009). HFACS has been demonstrated to have a high level of external validity (Shappell et al., 2007).

Design-induced error is a big concern of airworthiness authorities, especially in new, highly automated aircraft. Accident investigations of most major accidents have traced

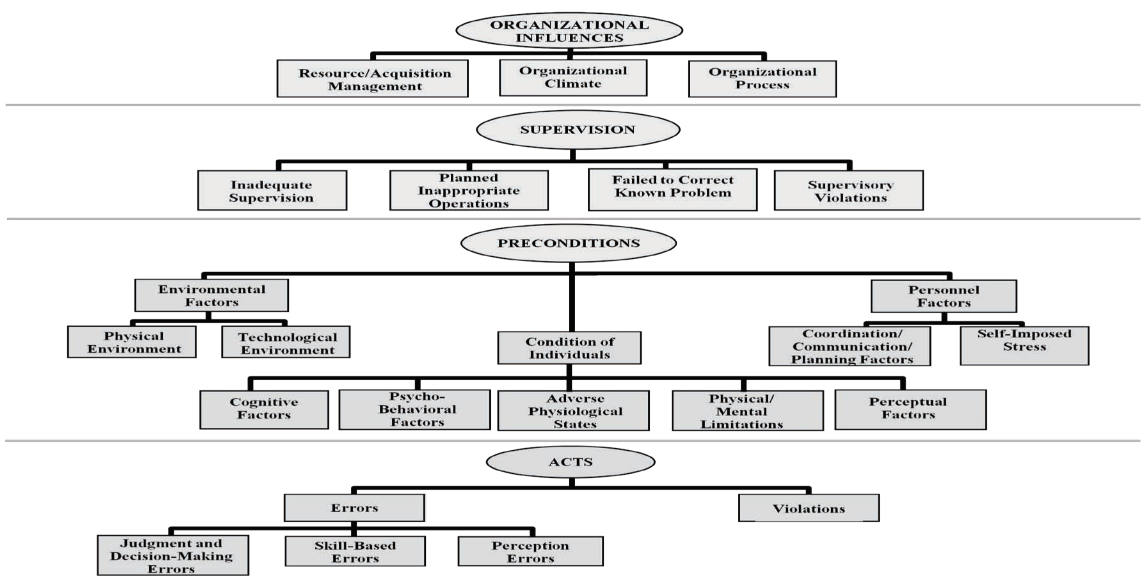


Figure 11. DoD Human Factors Analysis and Classification System (AFSC, 2007)

contributing factors to latent failures in system design (Salvendy, 2006). This led to a Federal Aviation Administration (FAA) recommendation that new aircraft designs should be evaluated for susceptibility to design-induced errors (Stanton et al., 2006). Attempts at doing this empirically are costly and must be done late in system development when the design decisions cannot be undone without great expense. Usability inspection methods, such as cognitive walkthrough, are based on human cognition models and attempt to address design problems earlier and cheaper (Bisantz & Burns, 2009). Other approaches for the early identification of design induced errors include the Systematic Human Error Reduction and Prediction Approach (SHERPA), Human Error Hazard and Operability study (HAZOP), and Human Error In Systems Tool (HEIST). In general, these are used in conjunction with other human factors methods, such as a hierarchical task analysis, to examine what might go wrong based on the type of activity. Dekker reviews these human error analysis approaches in (2006). Many organizations with mature reliability engineering programs use Failure Modes, Effects, and Criticality Analysis (FMECA) processes. This includes the Criticality Analysis process to evaluate the risk associated with potential problems (SAE, 2000).

While researchers have advanced the diagnosticity of human error, there has been a lack of focus on using this knowledge predictively. Most error models tend to be theoretical and academic. Practitioners have resorted to developing error-management programs based on intuition rather than on theory and empirical data. In fact, Johnson states, “It is ironic that there has been so much research into human error analysis and yet so little attention has been paid to those who must apply the techniques.” (1999). He also notes a “lack of integration between contextual analysis and requirements analysis.” (1999) He cites a need

for methodological support to help practicing engineers take contextual factors into account during design.

The method presented here is intended to meet the needs of the practitioner involved in new system development. It describes how to perform requirements elicitation early in the design process by performing an empirical study of legacy system mishaps involving human error as a causal factor. This is similar to the failure analysis methods used by the designers of aircraft structure or propulsion systems. They have established that successful preventative actions are based on a correct understanding of causal factors (Wiegmann & Shappell, 2003a). The White House Commission on Aviation Safety and Security made a definitive conclusion that the incidents of mishaps caused by human error can be reduced by the effective sharing of safety data (1997). This conclusion was reinforced by the recent crash of a US Air Force transport aircraft when it was found that the same adverse human-machine interaction had been cited in the crash of a bomber aircraft two years prior (Rolfsen, 2009).

Method

The proposed method involves a quantitative analysis of the significance of active and latent human component issues in similar contexts with similar systems to the system under development. Figure 12 shows a representation of the engineering process of investigating and preventing accidents that was first delineated in (Wiegmann & Shappell, 2003a). In the acquisition stage, a new system is developed. During this time, the system is also being inadvertently implanted with latent conditions for failure, whereby a hazard can manifest as a full blown accident in the fielded system. When an accident does occur, it is investigated

following mature and objective techniques and procedures. The findings are then classified and stored in a structured database. For human error incidents, this classification makes use of HFACS. This supplies accident analyses with useful and objective information, and forms the feedback to reduce accidents.

Mitigation efforts seek to reduce the incidence of similar accidents in existing systems. Mitigation efforts alone are insufficient. Prevention efforts must improve so that current operational difficulties are addressed in new systems. To do this, methods must contribute to removing the discontinuity between users and designers. Many advocate simply asking the users what should be fixed. This has proven unreliable, as demonstrated by an avionics evaluation by Newman and Anderson. In that study, the users consistently showed preference to the system with which they had more experience, even in light of demonstrably poorer performance (Newman & Anderson, 1994).

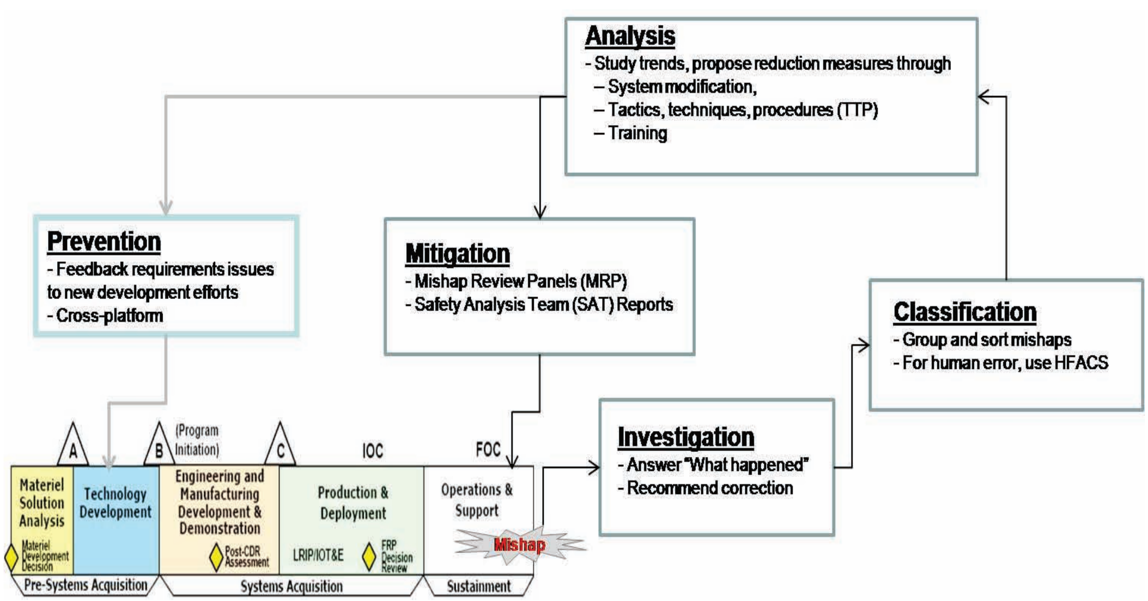


Figure 12. Engineering Process for Investigating and Preventing Accidents

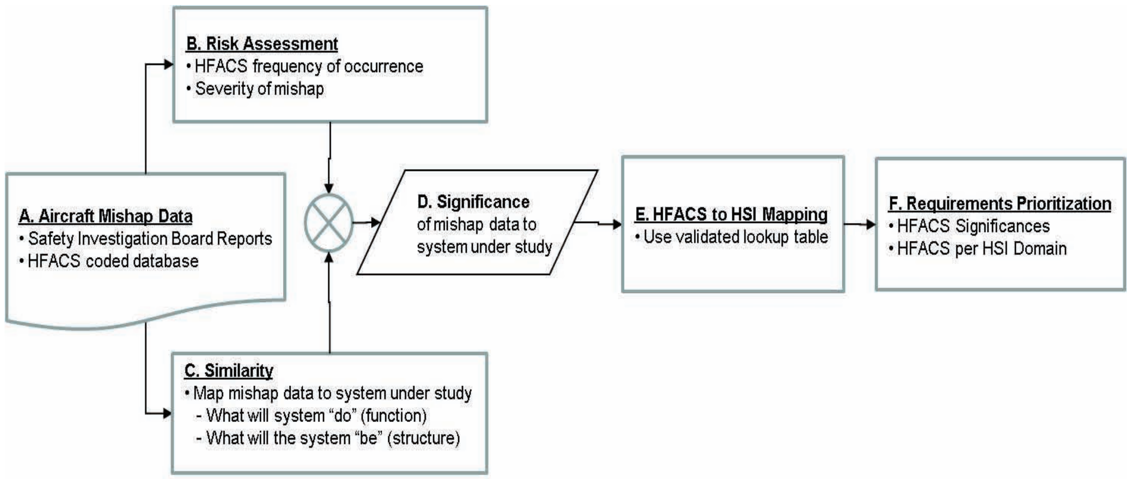


Figure 13. Process for Requirements Elicitation from Legacy System Mishaps

This new method contributes to a more robust accident prevention feedback loop. It also systematically maps issues to the domains of interest in DoD acquisition. This directly relates the requirements development process to the empirical data of the safety community and makes it more objective than would otherwise be possible. The method involves the following steps as shown in Figure 13 and expanded in the following sections.

Data Acquisition

The first step is gathering the proper data. Each branch of the DoD, along with most NATO forces, has an independent safety organization responsible for accident investigation and reporting. In the US Air Force, it is the Air Force Safety Center (AFSC) whose Safety Investigation Board (SIB) process is guided by AF Instruction 91-204 (USAF, 2006). As mentioned, the use of HFACS coding of human error is now mandated. SIB reports have limited releasability. All recipients must demonstrate legitimate need of the data and sign a nondisclosure agreement. However, derived conclusions are releasable as long as the data is presented in aggregate (e.g., no individually identifying data).

Risk Measurement

The data gathered by accident investigation boards on legacy systems can reveal valuable insight for the design of the next generation of systems, but only if properly analyzed to put it in useable form. The first step of making this accident investigation data usable is to quantify the level of risk that the identified issues present. The safety community analyzes hazards based on the probability of occurrence and the likely severity should an accident occur.

For DoD mishaps, the severity is determined by the class of accident as defined in DoD Instruction 6055.7. These are compatible with MIL-STD-882D suggested mishap severity categories. The values are quantified as shown in Table 2. The composite score of any one issue is determined by average severity level of the mishaps in which the respective issue was identified.

Table 2. Severity Classification

MIL-STD-882D			**DODI 6055.07**		**Quantified**	
Category	**Level**	**Summarized Criteria***	**Mishap Class**	**Summarized Criteria**	**Floor Class**	**Value**
Catastrophic	I	> $1M, death or perm. total injury	A	> $1M, death or perm. total injury, loss of a/c	A	**3**
Critical	II	> $200K, perm. partial injury	B	> $200K, perm. partial injury	B	**2**
Marginal	III	> $10K, non-perm. partial injury	C	> $20K, non-perm. partial injury	C	**1**
Negligible	IV	< $10K, minor medical	--		<	**0**

**Note: Rev 1 (currently in coordination) will raise these dollar amounts*

The probability analysis uses the mishap probability levels listed in AFI 91-301 which states expected occurrence on a per flight hour basis. This is compatible with the suggested probability values delineated in MIL-STD-882D. The method is applicable to any system safety program if the correct probability values are used. For instance, the values for risk

severity could be replaced by those used in the FAA System Safety Handbook and the method would apply to civilian mishaps. The values are quantified as shown in Table 3.

Table 3. Frequency Classification

MIL-STD-882D			**Specific System Safety Program Plan (IAW AFI 91-301)**		**Quantified**		
Category	**Level**	**Probability of occurrence**	**Category**	**Probability of occurrence**	**Floor per flt hr**	**~5 yr rate**	**Value**
Frequent	A	$> 10^{-1}$ (per FY)	Frequent	$> 10^{-4}$ (per flt hr)	10^{-4}	950	4
Probable	B	$> 10^{-2}$ (per FY)	Probable	$> 10^{-5}$ (per flt hr)	10^{-5}	95	3
Occasional	C	$> 10^{-3}$ (per FY)	Occasional	$> 10^{-6}$ (per flt hr)	10^{-6}	9.5	2
Remote	D	$> 10^{-6}$ (per FY)	Remote	$> 10^{-7}$ (per flt hr)	10^{-7}	0.95	1
Improbable	E	$< 10^{-6}$ (per FY)	Improbable	$< 10^{-7}$ (per flt hr)	$<$		0

The analysis of HFACS codes by severity can be very enlightening. For example, Wiegmann and Shappell found that judgment errors are more often associated with major accidents while procedural and execution errors more likely with minor accidents (Wiegmann & Shappell, 2003a). This correlation implies that the severity of a mishap is not simply a function of bad luck and shows the potential value of such analysis.

Our risk rating matrix, shown in Figure 14, is consistent with the MIL-STD 882D format. Using programmatic data it is a simple transform to express it in the format used by

Frequency of Mishap \ Severity of Mishap	*Negligible*	*Marginal*	*Critical*	*Catastrophic*
Frequent	0	4	8	12
Probable	0	3	6	9
Occasional	0	2	4	6
Remote	0	1	2	3
Improbable	0	0	0	0

Figure 14. Risk Matrix

the Risk Management Guide for DoD Acquisition (DoD, 2006) or the FAA System Safety Handbook (FAA, 2000).

Similarity Measurement

In our method, each mishap is also given a similarity weighting based on commonality with the proposed system. This is a novel component of mishap analysis, but increases the relevance of the resultant data for developers. Our similarity factor matrix, shown in Figure 15, captures the expected similarity of operator-vehicle interactions as the product of two dimensions: activity similarity and system similarity. This is sometimes referred to as the "Do-Be" weighting. Unlike the risk measurement, the similarity measurement must be tailored to a specific system under study.

The activity similarity rating defines the relationship between activities where mishaps occurred and activities to be performed by the system under study. These are quantified at three levels as shown in Table 4. The first is the broad class. For aircraft, this means all mishaps that were aviation related are at least this similar. The second and third levels are

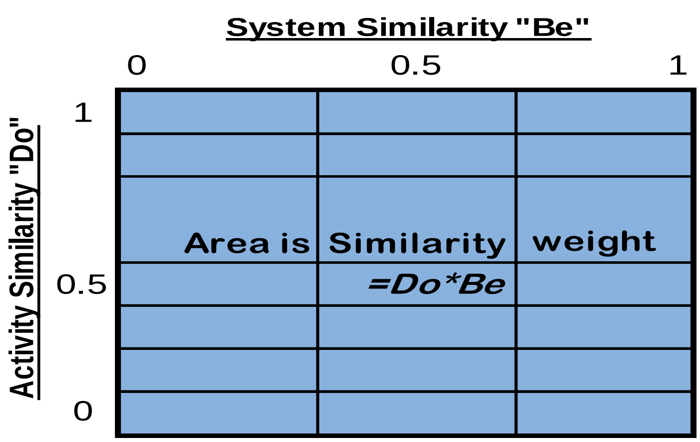


Figure 15. Similarity Matrix

defined in Table 5 as the general activities and their respective detailed categories. This list covers all activities identified in AFSC mishap investigations.

Table 4. Activity Similarity Classification

Operational	Contribution	Similarity Weighting	Notes
Broad Class	0.682	0.682	1 std deviation, Normal
Activity, general	0.272	0.954	2 std deviation, Normal
Activity, detailed	0.042	0.996	3 std deviation, Normal

Table 5. Activity List

Activity, general	Activity, detailed
Ground Operations	Maintenance
	Crew actions
Takeoff	Runway
	Carrier
	Austere
	Helicopter/Vertical
Landing	Runway
	Carrier
	Austere
	Helicopter/Vertical
Aerial Refueling	Provide fuel
	Receive fuel
Ground Attack	Direct engage
	Bomb release
Aerial Combat	Close range
	Extended range
Cruise	ATC/Navigation
	Night/Weather
	High altitude
	Low level
	Emergency/Unplanned event
	Formation
	Surveillance ops
	Airdrop
Acquisition/Development	(Policy/processes)*

**Note: Acq/Development, Policy/processes used only if HFACS code OR004 was sole finding*

The system similarity rating compares the system characteristics of those involved in mishaps and the system under study. These are quantified at three levels shown in Table 6. The first is the broad vehicle class. For aircraft, this means all heavier-than-air powered aircraft. The second and third levels are more specific. For military mishaps, the second level of system similarity is the model/mission design series (MDS). An MDS is the symbolic designation of aircraft type and model. This follows AFI 16-401, AR 70-50, and NAVAIRINST 13100.16 for the Departments of Air Force, Army, and Navy, respectively. A mishap aircraft reaches the third level of system similarity if it is the same, or a direct replacement for, the system under study. For future systems, this is would be the designated replacement which is a match in crew configuration, performance, and design.

Table 6. System Similarity Classification

Physical	Contribution	Similarity Weighting	Notes
Vehicle Class	0.682	0.682	1 std deviation, Normal
MDS Class	0.272	0.954	2 std deviation, Normal
Same Weapon System	0.042	0.996	3 std deviation, Normal

The similarity weighting of a mishap is the product of the similarity for both the involved activity and system. Each HFACS code is then given a similarity factor which is the average similarity weighting of all instances.

Significance

HFACS codes are each assigned a value that is their average risk measurement weighted by their similarity factor. We have named this parameter the *significance*. Recent studies indicate the need for this step in mishap analysis. The 2008 AFSC annual mishap report compared manned and unmanned aircraft. This juxtaposition revealed that none of the top

four causes of mishaps are the same for manned aircraft mishaps and UAV mishaps (AFSC, 2009). This implies that the occurrence of specific human errors is a function of the system characteristics.

Mapping to the domains of human systems integration

The final step is to express these findings in the categorization of the acquisition community. Program managers and systems engineers study the human components of their systems as part of a human systems integration (HSI) plan (DoD, 2008). HSI is normally expressed through its domains. Figure 16 shows the nine domains recognized by the DoD. Mapping HFACS codes into these domains is not just a bridge between two lists, but two separate paradigms. Mishap investigators use HFACS to study an event that occurred with an existing system; they want to capture the causes of the accident. Systems engineers increasingly use HSI domains as part of development (or upgrades) to create a system that does not yet exit; they want to know the important contributions to developmental success. We performed interviews with veteran systems engineers and

Figure 16. HSI Domains
(AIRPRINT, 2004)

program managers to get their views on where each of the 147 DoD-HFACS codes fit in the HSI domains. Table 7 lists the criteria used in that research. For the complete mapping, refer to (Hardman & Colombi, 2009).

Table 7. Considerations for Mapping HFACS to HSI

Domain	Related Mishap Factor	Questions for HFACS inclusion in Domain
Manpower	Workload, team composition	Were sufficient and correct people available?
Personnel	Necessary KSA*s	Did users have the necessary KSAs?
Training	Acquired KSAs	Were users sufficiently prepared?
Human Factors	UI*, work support systems	Did equipment support user in completion of tasks?
Safety	Hazards, warning systems	Were known hazards sufficiently managed?
Health	Injury/illness conditions	Were conditions causing injury/illness prevented?
Survivability	--	--HFACS does not cover intentional attack--
Habitability	Long term degradation	Were conditions that hurt performance prevented?
Environment	--	--HFACS does not cover effect on environment--

**Note: KSAs—Knowledge, skills, and abilities, UI—User interface*

Application

To demonstrate this new method, we apply it to design decisions for a new multi-role unmanned aerial system (UAS) called the MQ-X. While the method can readily apply to a broad range of analyses, the unique challenges of UAS design highlight the advantages of the proposed methodology over current practice. The MQ-X is planned to have the capability to transition commercial airspace, give and receive aerial refueling, and perform surveillance, reconnaissance, close air support, and strategic strike (Bowman, 2009). The research team developed a spreadsheet-based tool to implement the method. The tool is called RELAAy (Requirements Elicitation through Legacy Accident Analysis) and was built using Microsoft® Visual Basic® for Applications (VBA). It is available for download for approved recipients.

Data Acquisition

The data used for this study was all US Air Force aviation-related mishaps in which HFACS was identified as a contributory or causal factor between fiscal year (FY) 2004 and FY 2008. The data range begins in FY 2004 because that is the earliest mishaps that were coded with the current HFACS. The data ends with FY 2008 because that is the latest in which all mishap investigations have been completed. This data was analyzed using the RELAAy tool. As new data becomes available, the RELAAy tool can be updated using it's built in update function. Within this date range, AFSC cited 902 HFACS issues accounting for 207 mishaps. Of the 147 DoD HFACS codes, 120 were cited at least once in the five year span.

Risk Measurement

Risk measurement is independent of the system under study. The RELAAy tool contains this data for all mishaps within the data range. It was derived as described in the method. The first finding of interest is that mishaps where HFACS codes were involved were found to be over six times more severe, measured in cost of damage, than the average for all mishaps during the same time period (AFSC, 2009).

Similarity Measurement

Unlike risk measurement, similarity measurement must be tailored specifically for the system under study. For the MQ-X, the activity similarity weights were assigned as shown in Table 8. The proposed MQ-X activity information was obtained from the system description documents (Bowman, 2009). The assigned values are consistent with the weighting scheme presented in the method section, which is the default for the RELAAy tool.

Table 8. Activity List, MQ-X

Activity, general	Activity, detailed	MQ-X
Ground Operations	Maintenance	0.996
	Crew actions	0.996
Takeoff	Runway	0.996
	Carrier	0.954
	Austere	0.996
	Helicopter/Vertical	0.954
Landing	Runway	0.996
	Carrier	0.954
	Austere	0.996
	Helicopter/Vertical	0.954
Aerial Refueling	Provide fuel	0.996
	Receive fuel	0.996
Ground Attack	Direct engage	0.996
	Bomb release	0.996
Aerial Combat	Close range	0.682
	Extended range	0.682
Cruise	ATC/Navigation	0.996
	Night/Weather	0.996
	High altitude	0.954
	Low level	0.954
	Emergency/Unplanned event	0.996
	Formation	0.996
	Surveillance ops	0.996
	Airdrop	0.954
Acquisition/Development	(Policy/processes)*	0.996

Significance

The significance of each HFACS codes to the MQ-X program is derived from the risk measurement of the code and the similarity measurements of the involved aircraft as described in the method. Figure 17 shows the RELAAy tool analysis output. This is the distribution of significance of the respective DoD-HFACS codes to the MQ-X development program. As it can be seen by using the legend, RELAAy also uses the mapping to HSI to express the distribution of the codes among the HSI domains.

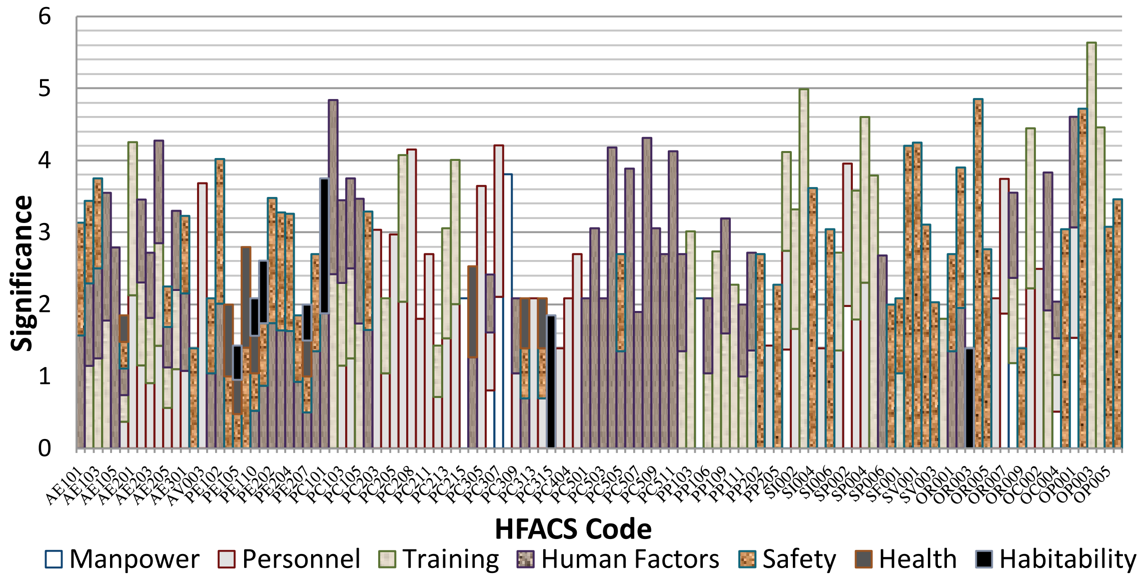


Figure 17. Program Overview, MQ-X Study

Requirements Prioritization

Table 9 lists the issues identified as most significant for the program; i.e. the top ten peaks from Figure 17. The general trend of Figure 17 is consistent with more generic UAS mishap studies such as (Tvaryanas, 2006b), but Table 9 reveals the considerable effect that the similarity factor makes in the prioritization of issues. We found that half of the top ten causal factors identified by the tailored similarity measurement would not have been identified by a generic (i.e., not tailored for similarity) risk measurement. Work is currently being done to validate this prioritization and to propose improved weighting schemes.

Table 9. Top HSI Requirements, System under study: MQ-X

HFACS	Description	Related Domain(s)
OP003	Procedural Guidance/Publications	Training
SI003	Local Training Issues/Programs	Training
OR004	Acquisition Polices/Design Processes	Safety
PC102	Channelized Attention	Training, Human Factors
OP002	Program and Policy Risk Assessment	Safety
OP001	Ops Tempo/Workload	Human Factors, Manpower, Personnel
SP004	Limited Total Experience	Training, Personnel
OP004	Organizational Training Issues/Programs	Training
OC001	Unit/Organizational Values/Culture	Training, Personnel
PC508	Spatial Disorientation 1 Unrecognized	Human Factors

Engineers with subject matter expertise in various domains were queried regarding the results. The consensus was that these findings expressed what their experience told them to be true. For example, one engineer with a background in training system development said of the prioritized requirements, "That figures! The pubs, guidance, and local training programs are always the last to get funded and the last to be planned for." (Wirthlin, 2009) Another said, "This method could finally give answers when the PM (program manager) wants to know the necessary representation for HSI on integrated product teams." (Mueller, 2009).

Conclusion

Researchers have established the importance of designing systems with the human components in mind. To do this, engineers must elicit and prioritize requirements related to the human components of the system early in system development. This new method is an empirical study of legacy system mishaps involving human error as a causal factor. It enables a thorough review of the mishap data in context to the activity and form of a system under study. By applying the similarity weighting and mapping to HSI domains, we can begin to bridge the work of the safety community with the systems engineering processes.

With updates to the human error taxonomies and mishap data, this method can remain relevant for the complex human-machine interaction of the aerospace industry.

V. Method 2: Empirical Function Allocation[2]

In this chapter, the methodology presented in Chapter 3 is applied to the logical analysis process of a prototypical acquisition program. It is shown that the use of the proposed method supports quantifiable design decisions.

Introduction

Though function allocation is critical in new system development, it remains insufficiently supported by quantitative methods. A function is a logical unit of behavior of a system (Blanchard & Fabrycky, 2006), and function allocation involves matching system functions, actions, and decisions with hardware, software, humans, or some combination of them (NASA, 2007). New system development efforts are seldom a blank slate; they often have function allocation constraints due to budget or compatibility considerations. This often leaves the designer with a limited number of explicit allocation decisions; however, these decisions have far reaching influence over the entire system architecture. Function allocation must be done in a manner that correctly balances cost, schedule, and performance with an acceptable level of safety. If one uses empirical data for function allocation decisions, those decisions will be more objective and will enable easier adaptation of new missions or technology.

A significant subset of function allocation concerns the distribution between humans and computers; that is, automation analysis. Automation is defined as the execution by a

---

[2] The contents of this chapter have been independently submitted for publication by Hardman, Colombi, Jacques, Hill, and Miller in the International Journal of Applied Aviation Studies.

machine agent of a function previously carried out by a human (Scerbo, 1996). It is difficult, but possible, to quantify the effect of automation on cost and schedule requirements. The designer can measure the expected effect of automation on operational efficiency and expense, and compare that with the expected penalties of higher equipment costs, more complex integration, and longer testing schedules. The alternative option, manual execution, consequently places more of the cost and schedule burden on manpower, personnel, training, and human factors.

The effect of automation on performance requirements is also complex; it is a classic measurement problem. Deciding how to deem a particular allocation *better* requires contextual data (Finkelstein, 2003). Is *better* a measure of precision--the computer's strong point, or resilience given unexpected conditions--the human's strong point? Many efforts to make automation decisions more scientific are influenced by the famous Fitts List which made axiomatic statements of what computers did better versus what humans did better (Fitts, 1951). This approach is problematic for two reasons: First, any static division is vulnerable to being rendered obsolete by evolving technology or changing system context. Second, and more significantly, this approach presupposes that all functions must be discretely allocated to either man or machine. Mitchell has reviewed other approaches for function allocation and cites that most are little more than design maxims. That is, they state generally accepted truths but are void of methods to guide their application (Mitchell, 1999).

We demonstrate a novel method to make the decisions of function allocation between human operators and computers that is quantitative, considers system context, is adaptable, and can be applied across a spectrum of automation options. We then demonstrate this method in examining the issue of traffic collision avoidance for unmanned aircraft.

Method

This method is intended to be performed in the framework of a comprehensive systems engineering effort. As portrayed in Figure 18, function allocation, as part of the Logical Analysis process, is the nexus between two iterative loops; the requirements loop and the design loop (CSE, 2008b). Therefore, function allocation decisions are made using preliminary system requirements and draft architectures. The decisions, in turn, determine interface design. In the end, validation testing must evaluate requirements for performance, physical characteristics, interoperability, human factors, and supportability (DoD, 2008). This method involves the following six steps.

1. The first step is to classify the lowest level functions of the functional decomposition as machine-only, human-only, or either. With computers becoming increasingly ubiquitous, a machine-only function has become a computer-controlled function. Examples include: demodulate transponder signal or charge battery. Examples of human-only functions are: express preferences or determine intent. These allocation decisions occur during the initial system architecting effort. Function allocation decisions for machine-only functions are

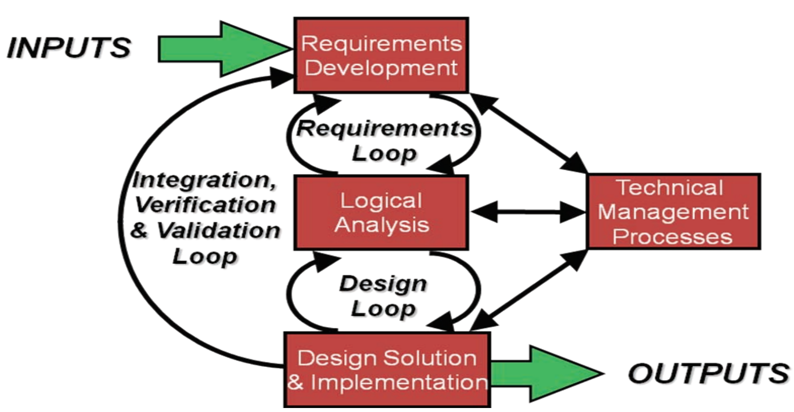


Figure 18. Systems Engineering Processes
(CSE, 2008)

relatively straightforward. They are based on such issues as the choice of distributed or hierarchical design and the details of the network structure.

2. For functions that indicate human involvement, more analysis is necessary. The next step is to perform a preliminary task analysis on these functions. This gives a better understanding of the *what* before the *how* or *who* of function allocation. Task analysis is much more mature than methods of function allocation, and there are multiple accepted techniques. For a thorough review of these see (Stanton, Salmon, Walker, Baber, & Jenkins, 2005). For human-only functions, this data is useful for user interface design. The remaining functions, those in the either category, require an analysis for automation. To do this, the function must be further decomposed into information processing stages. One well supported nomenclature for these stages is: sensor, processor, decision-maker, and actuator (Parasuraman, Sheridan, & Wickens, 2000). The dynamic between human and computer for a function can be very different based on the allocation of these stages.

3. Next, designers must quantify performance criteria for each information processing stage of each function under study. This forms the threshold and objective performance requirements for automation analysis. This quantification must consider overall workload and an acceptable margin of safety. For complex scenarios, the use of established methods of cognitive task analysis will form a better picture of workload (Crandal & Klein, 2006). A test program on the composite system in realistic scenarios could be cost prohibitive. To avoid this, we quantify the critical technical parameters necessary to achieve success and use those values as performance requirements. Using these lower-level criteria reduces time and cost and makes laboratory and ground testing possible. There are several different performance models to compare performance. There is a good summary of this in (Rouse

& Rasmussen, 1981). For data-limited tasks (as compared to resource-limited tasks which involve a speed-accuracy tradeoff), engineers generally model the problem using signal detection theory. Wickens has theorized that this approach could be used to make statistical parameterizations for automation decisions (2002). Signal detection theory assumes that one can identify the threshold detection of a signal distinct from the decision to act on it. A receiver operator characteristic (ROC) curve analysis is useful for evaluating such performance. A ROC curve, such as is shown in Figure 19, is a plot of Type I errors (or errors of omission: i.e. "should act, but didn't") and Type II errors (or false positives: i.e. "acted when shouldn't"). The probability of Type II errors (β) is plotted on the abscissa axis against the reciprocal probability of Type I errors (1-α) plotted on the ordinate axis. Even if there is insufficient data to create a complete ROC curve, it is important that the performance criteria consider both the sensitivity and the specificity of execution.

4. The next step is to quantify the performance of humans for each task in the functional decomposition. These will be specific to each function and will vary under

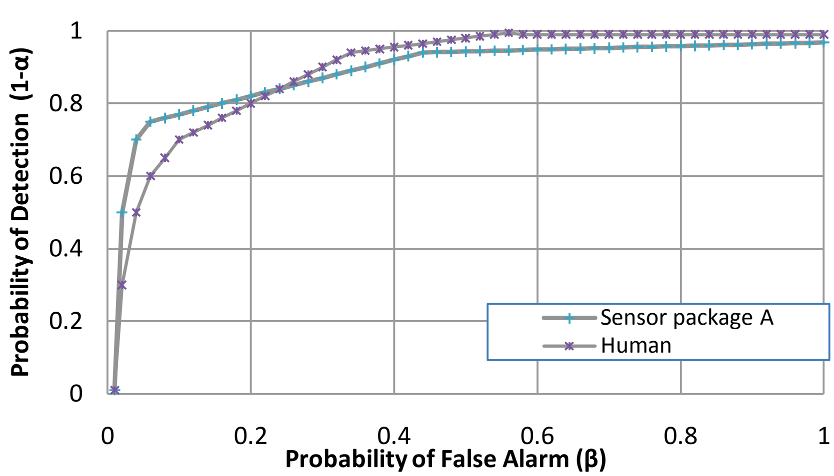


Figure 19. Sample ROC Curves for Automation Comparison

diverse real world conditions. Empirical data from basic research in physiology and psychology can be used to quantitatively characterize human capabilities and limitations. Though existent data is wide-ranging, this step may require additional, more specific, basic research.

5. Next, the performance of machines (sensors, processors, or actuators) for the same tasks is quantified. This data is obtained through trade studies and consultations with industry.

6. The final step is to compare the alternatives and make the allocations. In the past, designers approached automation as a way to eliminate the "inefficient and untrustworthy" presence of humans (Dixon & Wickens, 2006). This perspective fails to appreciate that the human components add necessary resilience for unexpected situations. Anyone that has ever observed a robot stuck in an unexpected situation can appreciate the need for resilience. A modern paradigm is that automation exists to augment human capabilities and to assist the operator in achieving system goals (Dixon & Wickens, 2006). The level of automation resides on a spectrum between the machine-only and human-only extrema. There are multiple sources for the delineation of this spectrum. Some texts use as many as 7 to 10 levels to define the range (Albery & Khomenko, 2003; Miller & Parasuraman, 2007; Parasuraman, Sheridan, & Wickens, 2000). The four levels of automation defined in operator role theory map to specific function allocation decisions. The role of the operator in each level is defined as: direct performer, manual controller, supervisory controls and executive controller (Folds, 1995). These are determined by the assignment of information processing stages between human and computer. Figure 20 shows this graphically. One

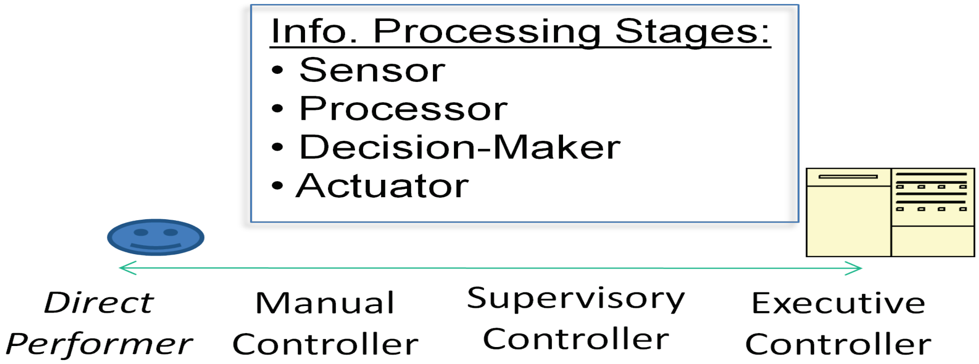


Figure 20. A Graphical Portrayal of Operator Role Theory

may also need to examine the need for adaptive automation. Adaptive automation may be necessary when operator workload is widely variant (Scerbo, 1996).

Application

We now use this method to guide design decisions for traffic collision avoidance in a new multi-role unmanned aerial system (UAS)[3]. While the method can readily apply to a broad range of systems, the unique challenges of UAS design make it a great example application to highlight the advantages of the proposed methodology over current practice. The new UAS, being considered by the US Air Force (USAF), must have the capability to transition through civil airspace[4]. A recent article in National Defense Magazine quoted the Director of the USAF UAS Task Force as stating, "national airspace integration is critical to international recognition of UAS flyover rights" (Jean, 2009). Industry officials state that the goal for UASs in the national airspace is an equivalent level of safety, including collision

[3] Unmanned aerial system (UAS) is the current official designation for what was known as a unmanned aerial vehicle (UAV) and its support components. When specifically the aircraft, or vehicle, is intended, UAV is appropriate. It must be noted that some are advocating a change to the term remotely piloted vehicle (RPV).
[4] Civil airspace is used to define that airspace under the control of a civil aviation authority, sometimes called the national airspace system (NAS), as compared to restricted or special use airspace.

avoidance, when compared to piloted aircraft (Warwick, 2004); however, the 2009 aircraft collision over the Hudson River in New York reignited the debate over the sufficiency of collision avoidance in even piloted operations (Baker & Grynbaum, 2009).

Function Classification

The first step of the method is to classify the functions regarding human and machine potential interaction. The regulations governing traffic deconfliction are contained within the "Right of Way Rules" section of the federal aviation regulations (FAA, 2005). By design, there are four layers that maintain safe operations in aviation. They are: airspace regulation, air traffic service (ATS), cooperative avoidance systems, and finally, see and avoid as the last line of defense.

<u>Airspace Regulation</u>

One of the primary ways that aviation authorities attempt to avoid traffic collisions is by defining airspace. Table 1 explains the airspace classes as designated by the International Civil Aviation Organization (ICAO). Though pilots are still required to practice see and avoid, the responsibility for conflict avoidance changes with the type of airspace. In addition, there is a speed limitation of 250 knots below 10,000' above sea level and 200 knots in Classes C & D (ICAO, 2001).

As Table 10 shows, safe UAV operation in Class B or C is technically possible and would only be necessary for short transitions. Due to the high traffic volume, however, there is likely to be continued opposition to granting widespread access to UAVs. Currently, UAVs are not allowed in any class of airspace without special permission. In the US, the FAA requires a certificate of authorization (COA) which requires at least a 30-day notice to

local administrators, visual meteorological conditions (VMC)[5], a route clear of all populated areas, and constant ground control by a certified pilot (FAA, 2002). Therefore, for regulatory reasons, route planning and airspace situational awareness must remain a human-only function. For example, the American military's Global Hawk UAV is given frequent approval to travel in Class A airspace over most of the world and with a minimum of pre-coordination, but the flight request must verify that it will first climb to altitude within restricted airspace (AF Tech, 2009).

Table 10. ICAO Airspace Designations and UAV Operations
(ICAO, 2001)

Class	Flight Ops	ATC equipment & services Provided	Radio Required	Transponder Required	UAV integration problems
A	IFR only	Radar Conflict resolution & separation	Yes	Yes	None *ACAS primary*
B	IFR/VFR By permission	Radar Conflict resolution & separation	Yes	Yes	No technical problem, bu traffic density increases risl *ACAS primary*
C	IFR/VFR After contact	Radar Separation (IFR), or traffic advisories (VFR)	Yes	Yes	No technical problem, bu traffic density increases risl *ACAS primary*
D	IFR/VFR After contact	Tower Separation (IFR only)	Yes	No	ACAS insufficient, *DSA primary*
E	IFR/VFR	Separation (IFR only)	Yes	No (<10,000´)	ACAS insufficient, *DSA primary*
F	IFR/VFR	Traffic advisories (IFR)	No (<10,000´	No (<10,000´)	ACAS insufficient, *DSA primary*
G	VFR	None	No (<10,000´	No (<10,000´)	Moderate problem due to lack of coverage, *DSA primary*

Notes: IFR—instrument flight rules, VFR—visual flight rules. ACAS—Automated collision avoidance system. DSA—detect, see and avoid.

[5] Instrument Flight Rules (IFR), Visual Flight Rules (VFR), Instrument Meteorological Conditions (IMC), and Visual Meteorological Conditions (VMC) are defined more completely in FAR 91.

Air Traffic System

The second layer of safety for all aviation is the air traffic service or air traffic system (ATS). It has traditionally been referred to as air traffic control (ATC), but its future name has been deemed air traffic management (ATM) to highlight the eventual evolution towards less controlling and more managing. The system involves a great deal of technology. In addition to the primary surveillance radars that detect all traffic by reflected radar energy, the secondary surveillance radars detect cooperating traffic transponder returns. Automated ATS systems analyze these radar returns to predict possible collisions. However, for the near future, this level will continue to involve human-only functions ranging from flight plan coordination and approval to traffic conflict intervention.

Many technical and regulatory changes are planned for the future ATM. One of these technical concepts involves inter-aircraft data links that would allow automated route deconfliction. These proposed future capabilities are analyzed in (Hardman, 2006). Unfortunately for UAVs, these changes do not appear to eliminate the need for independent deconfliction capability because: (a) participation in the deconfliction data links will not be universally mandatory and (b) all aircraft will still require a backup capability in the event of network failures.

Cooperative Traffic Avoidance

The third layer consists of onboard systems that cooperatively work to deconflict traffic. These systems are independent of, but compatible with, ATM systems. Such systems would enable automated deconfliction between aircraft. The current weakness of cooperative systems is the necessity for all aircraft involved to have a compatible functioning system. Multiple proposals are being explored for ways to add information on nonparticipatory

traffic. This has great potential as an independent automated traffic control technology, but for the near future, it cannot function as the sole source of deconfliction.

Detect, See and Avoid (DSA)

The last layer of safety is the independent ability for each aircraft to detect, see, and avoid (DSA)[6] other aircraft. In uncontrolled airspace, the inherent freedom means that all responsibility for separation lies with the pilot. In other airspace, the see and avoid principle is still required to be practiced to the maximum extent possible. This principle is not specifically mentioned in ICAO regulation, but it is described by the FAA as, "When weather conditions permit, regardless of whether an operation is conducted under IFR or VFR, vigilance shall be maintained by each person operating an aircraft so as to see and avoid other aircraft." (FAA, 2005). This regulation is satisfied in piloted aircraft by simply training the aircrew to perform a disciplined scan out the windshield (though more advanced aircraft also have augmenting technology, such as radar). DSA on a UAS is more complicated because any manual steps must be done remotely.

Preliminary Task Analysis

This second step is to perform a preliminary task analysis on the functions under study. For the UAS traffic avoidance scenario, the DSA function needs further examination. Since see and avoid is already performed by manned aircraft, this can be studied directly rather than predicted. We begin by stating the high level objective: *Provide traffic conflict information in sufficient time to prevent midair collisions*. The task analysis performed in (Hardman, 2006)

---

[6] Some sources use the alternate terms: sense-and-avoid (SAA), non-cooperative collision avoidance, or traffic deconfliction when referring to new systems. "See-and-avoid" is the primary term used in the regulatory sense. In this report the term DSA will be used throughout to mean any concept or system in effect for the primary purpose of preventing collisions between aircraft that have not been deconflicted by other means.

defined the basic tasks of DSA as the following steps. The information processing stage where this occurs is shown in parenthesis:

1. Scan field of regard (Sensor)
2. Detect traffic (Sensor)
3. Predict conflict (Processor)
4. Calculate feasible action (Processor)
5. Choose best action (Decision-Maker)
6. Execute chosen action (Actuator)

These tasks must all be performed continuously and simultaneously to satisfy the function of DSA. Current technology readiness and regulatory requirements prevent the last two steps from being fully automated. Therefore, the automation analysis is done regarding the sensor and processor stages.

Quantify Performance Criteria

Aviation regulations do not define the required level of DSA performance, but they have established see and avoid areas of responsibility (ICAO, 1962) as shown in Table 11. The DSA system must complete its detection, tracking, and predicting of an incursion with adequate time for an avoidance maneuver to be performed. For UAS operations, the time requirement must also include the relay time for any processing stages that are not performed onboard.

Table 11. Required Performance
(Multiple Sources, as listed)

Parameter	Required Performance	Source of Requirement
Time to Collision Warning	No value given Sufficient for a safe miss distance (>500'). Speed dependent	FAA's AIM, Ch 6, Sec 6 FAA-Order 8700.1
Detection Range	No value given Sufficient to achieve warning time requirement	N/A
Revisit Rate	Sufficient to achieve tracking within warning time requirement	N/A
Resolution	Sufficient to achieve tracking at required range.	N/A
Field of Regard (FOR)	+/-110° Azimuth +/-30° Elevation	(ICAO Annex 2) "Rules of the Air"
Traffic Volume	Sufficient for most crowded airspace (up to 12)	Derived from EUROCONROL website, statistics link

The required detection range (*R*) is a function of closure velocity (v_c) and necessary warning time (t_{c-}) as follows:

$$R = v_c \cdot t_{c-} \tag{1}$$

Closure velocity is a function of the angles in azimuth (Ψ) and elevation (Θ) between the two aircraft's velocity vectors (v_1, v_2). This is defined by:

$$v_c \ = \ {dR}/{dt} \ = (\ (v_1 \cdot \cos(\Psi_1) \cos(\Theta_1)) + ((\ v_2 \cdot \cos(\Psi_2) \cos(\Theta_2)) \tag{2}$$

Above 10,000´ there is no speed restriction except for those prohibiting supersonic flight. With only the Mach limitation, closure velocities can theoretically be over 1200 knots ground speed. However, at that altitude working transponders are required which means cooperative avoidance systems are capable of deconflicting traffic. As mentioned, aircraft are limited to 200 knots in airport areas and 250 knots elsewhere below 10,000´. This means aircraft below 10,000´ in Class E, F, or G airspace must be able to prevent collisions with closure velocities up to 500 knots (two aircraft, traveling head on, both going 250 knots).

Most UAVs and general aviation aircraft are speed limited but still could experience closure velocity of up to 400 knots (150 knots cruise speed, plus up to 250 knots for other traffic).

Necessary warning time includes both the time for the operator/processor to react and the aircraft to complete the avoidance maneuver. Engineers at the Spanish Aerospace Test Center (INTA) studied avoidance maneuvers for UAVs with various performance characteristics. In the vertical plane, most UAVs operate with too little excess thrust to perform a satisfactory abrupt zoom maneuver (rapid climb). This is true for most all small aircraft. A dive would yield the most rapid change in trajectory, but it is an undesirable option due to the effects of rapidly changing forces on the fuel system and payloads. Furthermore, unapproved changes in altitude while on an IFR flight plan may unsafely complicate the scenario for both controllers and operators. For these reasons, avoidance through a change in the horizontal plane is either necessary or at least preferred (Hardman, 2006).

Analysis was performed to calculate the time necessary to complete a horizontal plane maneuver. Figure 21 shows the geometry for the generalized worst case scenario. In this scenario, a warning was given at the minimum alert time based on a predicted collision at time (t_f) if the aircraft continued on flight path (b). The time (t_f) is measured from the conclusion of the warning and reaction time. We defined the minimum allowable miss distance (r_f) to be 500 feet based on the official definition of a near mid-air collision (FAA, 2009). The aircraft's actual trajectory (s) is based on a maximum rate coordinated roll to the desired bank angle (ϕ). For illustration a right turn is used, but it is done so without loss of generality. The solution is achieved using the Law of Cosines, the derivation of which is well

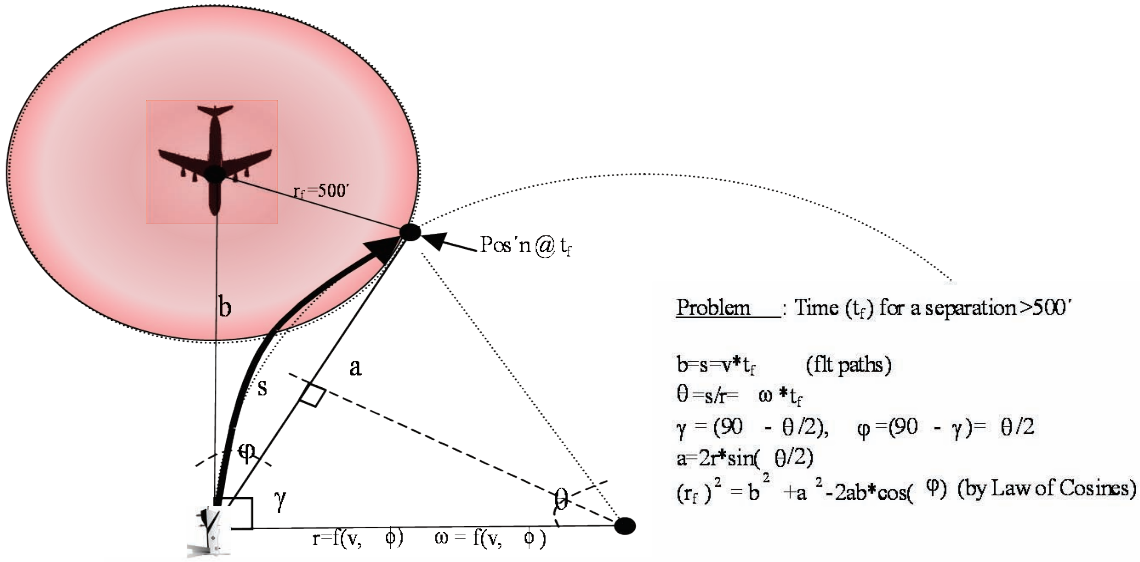


Figure 21. Avoidance Maneuver, General Solution.

established. The turn radius and the turn rate of an aircraft in level flight can be solved using the following established equations (USAF TPS, 2000b):

$$r = \frac{v^2}{g\sqrt{n^2 - 1}}; \quad \text{and} \quad \omega = \frac{g\sqrt{n^2 - 1}}{v} \tag{3, 4}$$

where:

v – velocity

g – force of gravity

n – load factor; which is equal to 1/cos(ϕ) in coordinated flight.

r – turn radius

ω – turn rate

Combining the equations of the generalized solution with equations 3 and 4 yields the following:

$$r_f^2 = (v\,t_f)^2 + \left(2r\sin\left(\frac{\omega t_f}{2}\right)\right)^2 - 2\left(2r\sin\left(\frac{\omega t_f}{2}\right)\right)(v\,t_f)\cos\left(\frac{\omega t_f}{2}\right) \quad (5)$$

with all variables defined as in Figure 21 and equations 3 and 4.

The INTA study sought to deduce the proper values for the variables in equation 5. It concluded that a maximum bank angle (ϕ) of 45° should be used for several reasons. First, many UAVs and small aircraft are limited in bank angle to 60°. A maximum rate turn should not be performed to the maximum allowable bank angle due to the consequences of overshooting the bank angle limit. Secondly, the short timeline limits the amount of time available to execute the maneuver. The study found that the UAVs were capable of roll rates between 20 and 30° per second; these are common values of normal small aircraft. At these roll rates, higher bank angles would require more time than allotted for the execution of the maneuver. Thirdly, as the bank angle increases beyond this value, the viewing geometry (for pilot or sensor) becomes a factor. At some angle, dependent on aircraft type and viewing position, it will not be possible to keep the traffic in sight throughout the turn. Finally, g-force and accelerated stall speed increase inversely to the cosine of ϕ which means the rate of increase becomes very high at high angles. Based on the above study, and using Equation 5 with a ϕ_{max} = 45°, the <u>necessary time to complete an avoidance maneuver is $t_f \geq$ 5.7 seconds</u>. If a different value for the maximum bank angle is desired, this value can be substituted in for ϕ in the equation 5.

One example of additional limitations is that satellite links frequently limit UAVs to ϕ= 15° during beyond LOS operations. The INTA study found that this should be programmed as a "soft stop", but, if necessary, the DSA system should be allowed to use the

maximum ϕ. The maneuver duration will be very short, so the link may not be lost. If it is lost, automated procedures for re-establishing the satellite connection are possible once the aircraft has returned to level flight.

A fact that is not intuitively obvious, but is implied by the preceding mathematical derivation, is that the warning time requirement is not speed dependent. As was shown, the required detection distance is proportional to the velocity; however, the turn rate is *inversely* proportional to velocity. In the time to collision calculations these two factors cancel out the speed dependence.

In addition to necessary warning time, aviation regulations define other performance requirements. It is necessary for a DSA system to provide coverage in the entire area of responsibility. Ideally, the system would provide +/-180° in azimuth (AZ) and +/-90 ° in elevation (EL); that is, total coverage. However, total coverage is not required. All aircraft are responsible for taking action to avoid traffic in an area consisting of +/-110° in azimuth and +/-30° in elevation. Additional coverage is desired from a "defensive driving" perspective, but the DSA capability must have a field of regard (FOR) at least this big. Also, any new DSA system must be able to cope with the highest traffic densities likely to be encountered. Modern computers have no problem exceeding this number of simultaneous predictions, but the system must be able to discern the highest threat at all times.

System integrity is a measure of how well the data can be trusted. For a DSA system it is primarily the probability of missed traffic, false alarms, or incorrect prioritization of intruders. The technical reasons for these problems include target location ambiguities, improper noise rejection, tracking ambiguity, and errors by predictive algorithms. Ideally, the system would have a 0% probability of missed traffic and a 0% false positive rate. This

is impossible to achieve much less evaluate. Aviation systems already have established threshold levels of safety that apply to these systems. If a DSA system is functioning as the primary method of separation, and traffic on a conflicting flight path is non-cooperative, then a missed or incorrect detection could lead to a midair collision. The FAA requires such catastrophic events to have a probability of occurrence of less than 10^{-9} events/flight hour (FAA, 2000). That is once every billion hours of operation.

Quantification of Human Performance

Governing agencies require UAVs to demonstrate an "equivalent level of safety" to that of manned aircraft (FAA, 2002). If an official quantitative definition of equivalent level of safety existed, then, airworthiness requirements could be directly derived from that definition. Unfortunately, no such definition has been endorsed by any regulating agency. The human's capability and limitations, when acting as the direct performer for DSA, have been determined by a meta-analysis of multiple human subjects studies. All parameters from the meta-analysis are listed in Table 12.

Table 12. DSA Human Ability
(Multiple Sources, as listed)

Parameter	Human Performance	Source
Time to Collision Warning	Needs greater than 18.2 s [1]	(BASI, 1991) & calculations contained herein
Detection Range	1.14 to 1.84 NM for 90% confidence	(Andrews, 1991; Hardman, 2006)
Revisit Rate	16 sec	FAA-P-8740-51
Resolution	0.3 mrad	(Smith, 2000)
Field of Regard (FOR)	+/-180° AZ +/-30° EL	N/A
Traffic Volume	Up to 5	FAA-P-8740-51

Note: 1: Derived from 12.5 s for pilot reaction and 5.7 s for avoidance maneuver (non fighter/aerobatic).

The most significant limitation is the required collision warning time. Research for this was originally performed by the Australian Traffic Safety Board, Civil Aviation Authority (CAA) formerly called BASI (BASI, 1991). Their results are generally accepted by the aviation community and have been cited in multiple accident investigations and subsequent research (Andre & Kukura, 2009). We summarize the results graphically in Figure 22.

It is alarming to note that these results indicate that for DSA the "equivalent level of safety" standard is actually insufficient to meet the FAA's necessary level of safety against catastrophic events. Using the optimal human performance values listed in Table 12 (R=1.84 NM and necessary t_{c-} = 18.2 sec), the maximum closure velocity (v_{c_max}) safely protected by human see and avoid is v_{c_max} = 364 knots. This does not take into account the human scan rate. The FAA recommends that pilots re-scan every 16 seconds. More frequent complete area scans are a worthy goal but difficult to achieve during high workload. At this recommended rate, the necessary detection time to prevent a collision is up to t_{c-} = 34.2 sec which equates to a maximum safe closure velocity of v_{c_max} = 194 knots; well below

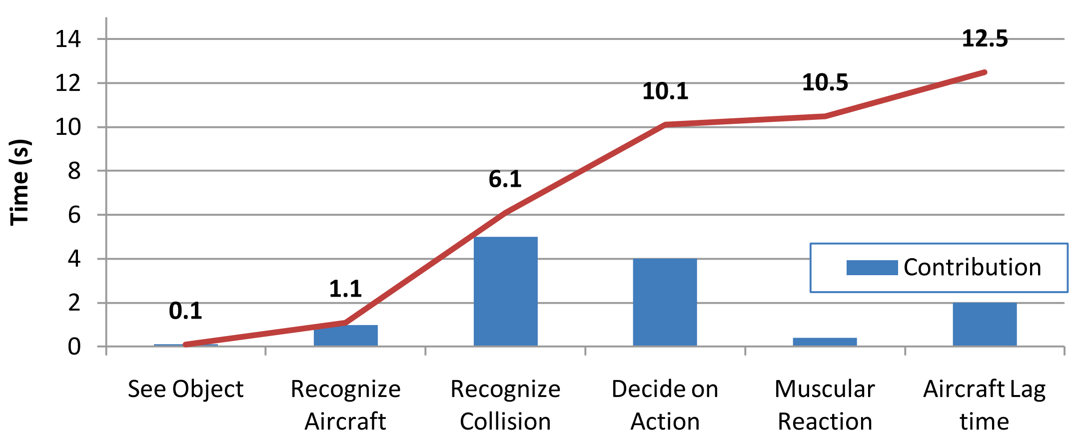


Figure 22. Time to React to a Collision Threat, Onboard Pilot

the speeds that aircraft can legally travel!

Quantification of Machine Performance

The next step is to quantify the potential performance of automated replacement. For automated DSA systems, detection performance is a function of revisit rate, detection range, and/or resolution. This requires the use of the governing equations for optical and radar-based technologies. For radar systems, the maximum Range (R_{max}) is given by (Stimson, 1998) as:

$$R_{\max} = \sqrt[4]{\frac{P_{avg} G \sigma A_e t_{ot}}{(4\pi)^2 S_{\min}}} \qquad \text{Radar Systems} \tag{6}$$

where:

P_{avg} – Average power

G – Antenna gain

σ – Target radar cross section (RCS)

A_e – Effective antenna area (Product of the physical area and an efficiency factor)

t_{ot} – Time on target, dwell time, or integration time

S_{min} – Minimum detectable signal energy

Except for RCS and dwell time, these parameters are all limitations of the physical system. Increases in range through increases in the power, gain, or area invariably come with consequence in weight, size, power, and money.

The size and weight of the system are dependent on its necessary gain, and gain is a function of the system's design wavelength. It is given by (Stimson, 1998) as:

$$G = \frac{(4\pi) A_e}{\lambda^2} \qquad \text{Radar Systems} \tag{7}$$

where:

λ – Wavelength

The wavelength is equal to the speed of light (c) divided by frequency (f); $\lambda = c/f$. Thus, the aperture size decreases or the gain increases proportionally to the square of the frequency. This makes radar systems operating at higher frequencies attractive options for UAV installation where payload is limited.

For the target RCS, most specifications use a 3 m^2. To effectively use the dwell time parameter, it is important to design a good scanning technique. The optimal dwell time is a tradeoff with revisit rate and field of regard (FOR). The longer one dwells in any one part of the sky, the longer it takes to view the total area of observation.

For infrared imaging systems, the range is related to angular resolution by (USAF TPS, 2000d):

$$R = W_R / \Theta_R \qquad \text{Infrared Systems} \tag{8}$$

where:

W_R – Linear resolution (minimum resolvable distance or diameter of target)

Θ_R – Angular resolution [rad]. The inverse of the spatial cut-off frequency ($f_{s, co}$)

For all imaging systems, the resolution is a function of the number of picture elements (pixels). However, high resolution imaging creates challenges for the processing system because of the large quantities of data that must be normalized and analyzed in real time.

For systems using laser technology the maximum detection range is determined by the required power (P_R) which can be determined by the laser range equation given by (USAF TPS, 2000c):

$$P_R = \frac{P_{XMTR} D^2 \rho_T}{4R^2} \eta_{ATM}{}^2 \cdot \eta_{XMTR} \cdot \eta_{RCVR} \qquad \text{Laser Systems} \tag{9}$$

where:

R – Range to target

P_{XMTR} -- Power in the transmission path of the laser

D – Detector aperture diameter

ρ_T – Target reflectivity

η_{ATM} – Transmissivity, atmospheric

η_{XMTR} – Transmissivity, transmission path of the laser

η_{RCVR} – Transmissivity, receiver path of the detector

Except for systems capable of simultaneous omni-directional monitoring, knowing the maximum range is not sufficient. After detection, the DSA system must track the traffic to determine if a potential for collision exists. This requires a minimum of three scans for accurate calculation (real world trajectories are arcs). The necessary range is also a function of revisit rate. It must be assumed that the traffic is just outside of maximum detection range in the previous scan. Thus, the distance the traffic can close before being detected and tracked is equal to the closure rate multiplied by the time the system takes to perform three complete scans ($3 \cdot t_r$). Substituting this into Equation 1 yields an actual time to collision of:

$$t_{c-} = \frac{R\text{-}(3t_r\ v_c)}{v_c} = \frac{R}{v_c}\ \text{-}3t_r \tag{10}$$

Predicted performance can be obtained from the given equations. The information necessary to use the equations is available from manufacturer data or applicable regulations.

To meet the derived requirements, numerous possibilities exist. There are potential tradeoffs between FOR, revisit rate, dwell time, and range (or resolution in the case of electro-optics). Thus, technologies which exceed the necessary performance in one area can make tradeoffs for improvement in others. For instance, longer dwell times can increases the detection range for radar systems. For electro-optical systems, an increase in sensitivity and a longer dwell time can increase the detection range. However, these longer dwell times increase the total area scan time. Therefore, revisit rate and maximum detection range are conflicting parameters of the time to collision requirement. These necessary tradeoffs are seldom discussed in product literature, but they are essential in helping the designer find feasible alternatives. Next, we examine the current state of the art in sensor technology and automation.

Visual Imaging

Visual imaging technology consists of using some form of camera arrangement to help establish situational awareness. It is the most analogous process to that of piloted see and avoid. Visual imaging is a semi-passive system in that no onboard illumination is needed for detection. Modern camera technology holds great possibility for small, low power cameras with very good zoom capabilities. In the future, a virtual reality system could theoretically give the remote operator the same visual scan as an onboard pilot, but the bandwidth and equipment requirements would not justify its use simply for DSA. A more realistic option is using image processing and a target recognition algorithm to analyze the input for the operator. The operator remains in the loop by cueing cameras to focus on places interest.

The primary weakness of these systems is that they are limited in the same way as the human vision. Some technologies can augment the picture for night and in haze, but

performance still suffers. Another inherent difficulty with electro-optical imaging systems is that, unlike radar, they do not have the capability to directly measure range. This is a big drawback for DSA systems as this is the primary parameter for calculating traffic avoidance. One possibility is stereoptic vision with sensors on the wing tips. This would use the same principle as the human brain to discern the distance of an object by simultaneously viewing it from two different angles. Unfortunately, this method is only effective at short distances. Beyond those distances the human being uses assessments of the apparent size of an object to determine distance (Physiological Training Office, 2001). A computer could do this as well, but it must know the actual size of the object. Range rate can be determined simply by measuring the rate of change in apparent size, but requires very high resolution systems as the apparent size of an object does not change rapidly until very close. A more probable solution to the range problem is to combine the electro-optical system with one of the technologies discussed later.

Infrared

The infrared (IR) region is lower in frequency (higher in wavelength) than the visible spectrum. IR technology uses the fact all objects radiate energy at a quantity proportional to their temperature, and aircraft have a temperature contrast with the surrounding sky. Unless used with an IR illuminator, these systems are passive; they use received energy only. IR systems operate in the electro-optical area of the spectrum, and so share many of the same properties and limitations of systems in the visible range. A typical IR system requires a signal to clutter ratio greater than 19 in order to achieve a 99% probability of detection (USAF TPS, 2000e). Though IR search and track (IRST) systems have been used by the military, there are currently no IR DSA systems in use. The only known proposal is a NASA

and US Navy effort to develop a supplementary IR-based DSA system with a proposed range of several miles and a FOR of +/-105° in azimuth and +/- 35° in elevation (Adams, 2001).

Laser Radar

Laser Radar technology operates in the visible and near-IR spectrum. A laser (light amplification through stimulated emission of radiation) is a system capable of generating an intense coherent beam of light. This beam is less susceptible to the atmospheric attenuation of other electro-optical systems. When reflected this beam can be sensed by a detector which can determine the distance of the reflected object. A laser detection and ranging (LADAR) system uses this feature to make accurate range measurements at long distances. Since the range is a part of each sensed beam, the system can provide a three dimensional perspective of the reflection. Some sources also used the term light detection and ranging (LIDAR) and include the use of ultraviolet lasers as well. The advantages and limitations of LADAR systems are both related to their very precise focused beam. LADAR systems provide high resolution in range and angle, but to cover a sufficient FOR they require very fast scanning, and real-time signal processing. There is currently no DSA system in development that makes exclusive use of LADAR or LIDAR; however, its ability to augment other systems is being explored (UAVM, 2009).

Radar

Radar (radio detection and ranging) systems have been used to detect aircraft since the 1940's. Like LADAR, which was derived from radar principles, Radar is an active system that sends strong pulses of energy and analyzes the returned signal. Two locations of interest in the radar area of the spectrum are at 35 and 94 GHz. The need for large external

apertures makes radar systems difficult to implement on small vehicles, and propeller-driven aircraft have a difficult time dealing with the interference issues that the propeller and engine can cause. Pusher propeller configurations allow for the installation of radar in the nose, but the size, weight, and power requirements make them currently unfeasible. Unlike the previously discussed technologies, some radar system has been developed and evaluated for DSA on UASs. Flight Safety Technologies is preparing to field a UAV version of its UNIversal Collision Obviation and reduced Near-miss (UNICORN™) system (Flight Safety Technologies, 2009), and there are reports that both Northrop Grumman and General Atomics are working on radar-based collision avoidance systems built specifically for installation on UAVs (UAVM, 2009). According to a Sandia National Laboratories press release, they believe that they are near the creation of a synthetic aperture radar that will have an effective range of over 7 NM and weigh less than 20 pounds (Sandia National Laboratories, 2004). Currently, the most mature system is the OASys (Obstacle Awareness System) radar installed on UAVs built by Scaled Composites, LLC (Wolfe, 2004). Designers set a range objective requirement of 6 NM. Initial NASA tests found the system was capable of detection ranges between 2.5 to 6.5 nautical miles, but there were some complete misses. Based on NASA´s flight test results, listed in Table 13, this system comes close to meeting all necessary performance requirements as stated. Regarding physical characteristics, the total weight is about 55 pounds and the externally mounted antenna is 16”x16”x22” (Wolfe, 2004).

Table 13. Evaluation of OASys Radar for DSA

Parameter	System Performance	Notes
Time to Collision based on: -- Detection Range & -- Revisit Rate	2.5 –6.5 NM 150 °/sec	Generally sufficient for UAVs
Resolution	1.7 mrad (0.097°)	Sufficient to achieve tracking within warning time requirement
Tracking Accuracy	Range: <5 m	Sufficient to achieve tracking at required range including worst case scenario ambiguities.
Field of Regard	Typical: AZ: ± 30°, EL: ± 11° Max: AZ: ± 90°, EL: +25° & - 85°	Less than the ICAO requirement.
Other	Altitude limitation of 20k´	Problem for most UAVs

Automated Processing

The sensor is just the first decision, but it influences the allocation of the other steps of automation. Tasks allocated for manual execution must be done remotely, but tasks allocated for automation must then be studied for the allocation of being performed onboard vs. on the ground. At this time, not fully automated onboard DSA has been fielded, even in prototype. Though such a capability would have many advantages over less independent alternatives, the numerous technical and regulatory challenges of such a system mean that it is not likely to be an option for many years.

We re-examined the decision timeline in light of a UAS scenario with a human operator acting as the manual controller. This is shown in Figure 23 and is analogous to the piloted aircraft but with added delay due to transmission. This transmission delay must be added to the total time twice, once for the alert and once for the control message. This transmission time delay consists of a propagation component and a relay processing component. Propagation time is the result of transmission range divided by the speed of light. This results in a time delay of 6.18 μsec per nautical mile. For this UAS analysis, one-way

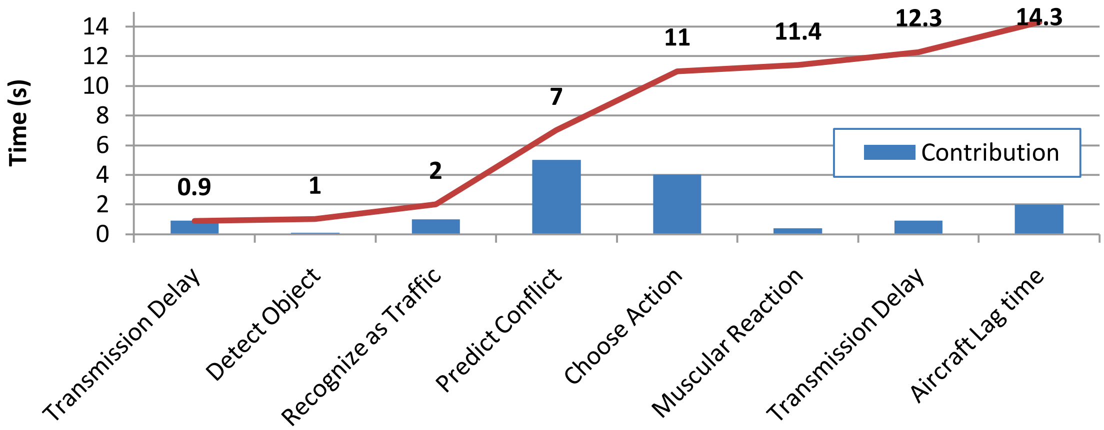


Figure 23. Time to React to a Collision Threat, Remotely Piloted

propagation time was estimated at 0.9 s, and relay processing of the message is assumed to be negligible. Attempting to satisfy the DSA function in this way results in a necessary detection range that is significantly farther than any proposed system expects to ever achieve. Thus, it is deemed infeasible.

The previous conclusions guide the search for a feasible solution toward a supervisory controller option. We re-examined the decision timeline in light of a UAS scenario that uses both onboard collision prediction software and remote human interaction. Such software has been developed and tested (Chamlou, Love, & Moody, 2008), and the human-in-the-loop fulfills regulatory requirements. As shown in Figure 24, this yields a necessary warning time that is more achievable. Based on these values, the UAS requirements for DSA are listed in Table 14.

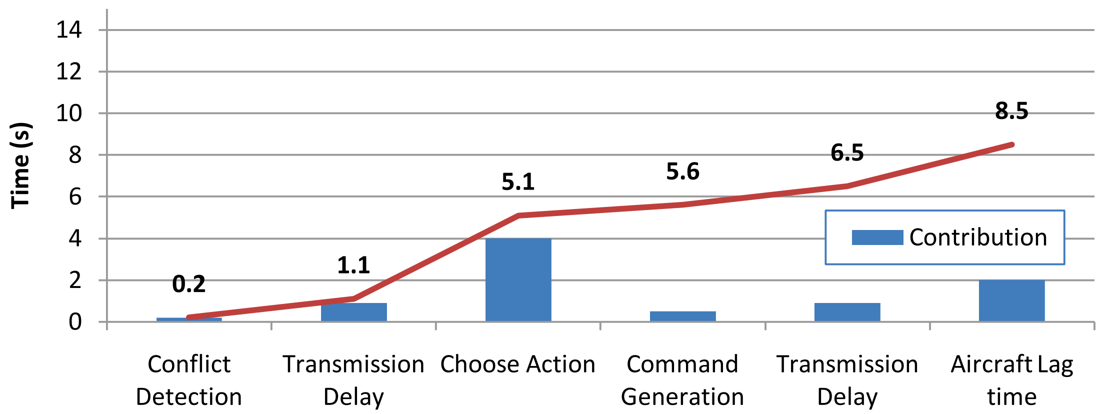


Figure 24. Time to React to a Collision Threat, Automated Detection

Table 14. DSA Necessary Performance of UAS Automation

Parameter	Required Performance	Source/Notes
Time to Collision Warning	Unlimited 14.2 sec Speed limited 12.2 sec	Sufficient for a safe miss distance (>500') below 10,000´. Value is for head-on traffic. Less for off angle traffic.
Detection Range & Revisit Rate	Minimum to achieve time requirements above: Unlimited 2.0 NM Speed limited 1.4 NM	Sufficient to achieve tracking within warning time requirement.
Resolution	--	Sufficient to achieve tracking at required range including worst case scenario ambiguities.
Field of Regard	+/-110° Azimuth +/-30° Elevation	Required performance: Those dictated in the ICAO "Right of Way rules". Desired performance: Total spherical area.
Traffic Volume	> 10	Busiest airspace densities for UAV operation

Note: the speed-limited category provides for a more easily attainable requirement for . Time includes transmission, reaction, maneuver, and propagation time.

Using equation 5, we see that an automated system with a revisit rate of 1 Hz could provide the necessary alert time to an operator if it had a detection range of 2.0 NM. This is within the predicted performance of at least one sensor under development.

Automation Selection

A comparison of the human and automatic requirements is made in Figure 25. The necessary warning times are plotted on the ordinate axis and the corresponding necessary detection ranges are plotted on the abscissa axis. The lines radiating from the origin are reference speed lines (V_c=200, 300, 400, and 500 kts respectively). Because the piloted aircraft equivalent level of safety is insufficient, UASs will ultimately have to gain certification by ICAO's alternative method; evaluation of system risks against a threshold. This method requires the advocating party to quantify the system performance and compare against an approved risk level (ICAO, 2001).

Based on the automation analysis above, and after an examination of the available sensor technology, there are currently no solutions for small UAVs that are fully satisfactory;

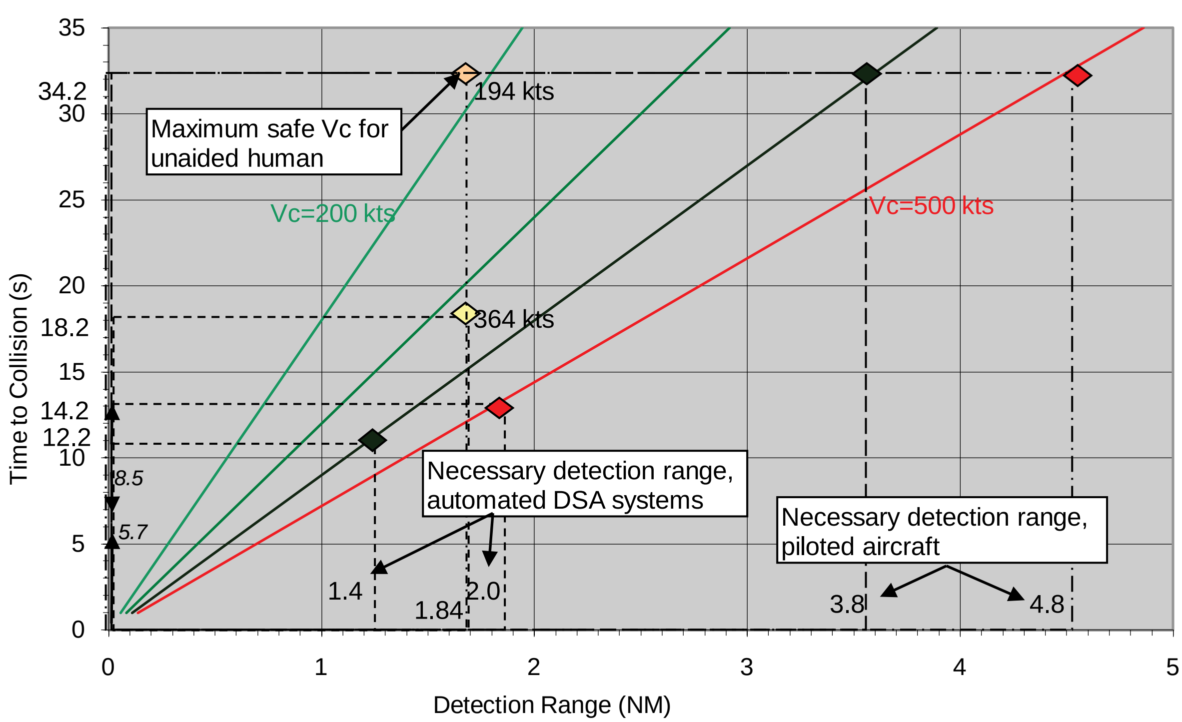


Figure 25. Human and Automated Detection Range and Time to Collision

though the use of a combined visual and LADAR system holds great promise for the future. Those UAVs will require some off board tracking and active deconfliction system, which means they will be limited in flexibility and dependent on another system. For larger UAVs that can handle the large expense and weight, onboard radar and conflict detection software is the best available option.

Conclusion

The method presented in this report is independent of the technology reviewed and can be used to perform quantitative function allocation between humans and computers. The results for the method, as applied to DSA in a UAS, show how automation analysis can be improved. In this particular application, the available technology makes for a difficult selection. The method made the choice clear and objective, and as these technologies mature, the analysis is easily updated to examine if the conclusions remain valid.

VI. Method 3: User Interface Design: Input Devices[7]

In this chapter, the methodology presented in Chapter 3 is applied to the design of input devices. It is shown that use of the proposed method improves predicted system performance.

Introduction

This method improves early system development by incorporating human component parameters in input device design. Interface design and management is an important part of systems engineering because it crosses traditional domains and is critical to safe and effective system performance. Engineers have been designing input and control devices for centuries, but in the abundant texts on design methods there is a conspicuous lack of human-inclusive techniques (Franklin, Powell, & Emami-Naeini, 1994; Johnson & Winters, 2005; Rerucha & Krupka, 2006). This method employs a multi-objective nonlinear optimization algorithm to find the input device controller gains based on the performance of a total system model; one that includes machine dynamics, the human operator, and the mission context. Rather than operating on a traditional closed form objective function, it employs a simulation-based approach. This allows it to accommodate real-world nonlinearities like empirical data for human capabilities and limitations where no characteristic equation exists. In addition, this approach can be executed early in the development process to provide better estimates for input device design. The model can then be progressively refined as the system matures.

[7] Some of this chapter has been published independently by Mathworks© as a MATLAB© User Community Application.

Many systems activities require manual tracking or selection tasks. These tasks require user interfaces (UI) that allow human operators to couple with machines; the operator knows the system goal and provides inputs to the machine (called the "plant" by control systems engineers) using an input device. Feedback is then obtained by the operator either directly from the environment or through a display device. Operator performance can be defined by the human perception, cognition, and neuromuscular capabilities and limitations. The tasks are often defined by the necessary tradeoff between speed and accuracy and in the presence of outside disturbances. Figure 26 depicts the functional block diagram of a system with a manual tracking task. The diagram uses the convention of control systems engineering to represent specific mathematical relationships, but the terminology is from human factors engineering and applied psychology.

One of the more complex examples of such a UI is an aircraft flight control system (FCS). Flight control systems are designed for favorable aircraft handling qualities and the avoidance of pilot-in-the-loop-oscillation (PIO) (DoD, 1990). The development and certification of an FCS involves sophisticated air data model formulation and complex design processes. For example, the USAF Test Pilot School delineates the following steps in

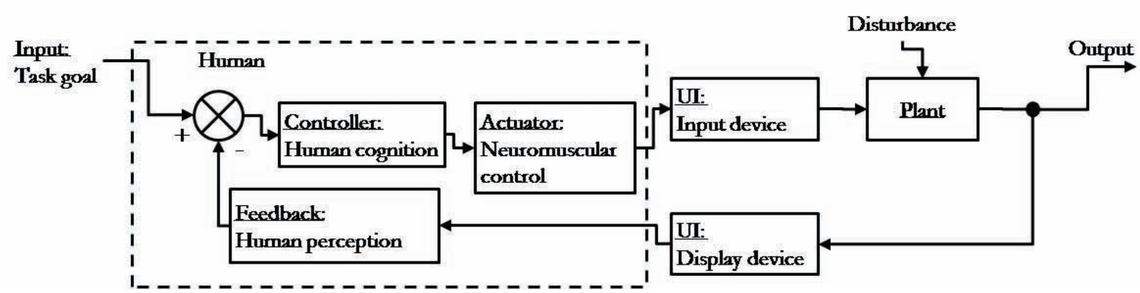


Figure 26. Block Diagram for User Interface (UI) Design

the FCS design process (USAF TPS, 2000a):

1. Characterize machine (sensor/processor) dynamics using wind tunnel data, computational fluid dynamics, and bench testing.
2. Create a high fidelity model of the system by using the above data to develop a closed form transfer functions of each component.
3. Define performance metrics for handling qualities and PIO susceptibility.
4. Determine derived requirements for system parameters from predictive criteria. For example: predicted handling qualities and PIO susceptibility can be inferred from the response to force applied curves and the aircraft transient response.
5. Determine the parameters of the flight control system that meets the requirements for system performance.

There is no straightforward analytic alternative for performing this process. It involves both computationally expensive analysis and significant manual parameter "tuning". This is due, in part, to the fact that the system requirements call for favorable pilot-machine interaction, but nothing in the model captures this interaction; it is only inferred from the predictive criteria of flying qualities. Furthermore, the solution must be refined by developmental test and evaluation in progressively higher fidelity media (i.e. simulator, ground, and flight test).

Other input devices are not nearly so complicated, but are still critical to system performance. These input devices must account for human capabilities and limitations as well as machine dynamics in tasks such as: object selection, robotic teleoperation, or targeting system slewing. A study of recently fielded systems determined that many of the systems' user interfaces are never designed based on proper performance metrics (Hoffman & Elm, 2006). The following method offers a straightforward and economical way to improve this part of system development.

Method

The following steps form a method for estimating input device parameters early in system development:

1. Characterize machine (sensor/processor) dynamics specific to each assigned function and under real world conditions. Perform the required trade studies to get data, if necessary.
2. Characterize human capabilities and limitations specific to each assigned function and under real world conditions. Get representative data from academic and research organizations, if necessary.
3. Model the system using the characteristics determined in the previous two steps. A closed form mathematical model is seldom possible, but numerous simulation software packages exist to model the nonlinearities of both human and machine.
4. Define context-based performance metrics and bounds of the system.
5. Formulate a simulation-based optimization problem. The objective function will be determined by the performance metrics, the constraints will be determined by the system bounds, and the design variables will be the modifiable parameters of the input device.
6. Run the optimization. The simulation will provide a predicted system response which must then be transformed into a form usable by the optimization algorithm.
7. When the optimization algorithm converges or terminates, the results will provide a best estimate of the input device design parameters.

The implementation of this method is demonstrated for a prototypical system using a tool coded in the Matlab© programming language and a model designed in the Simulink® simulation environment.

Application

This method is applied to the generic human-in-the-loop scenario portrayed in Figure 27. The input device must assist the human operator perform manual input tasks such as those described previously. The task is complete when the system state (the output) matches the task goal (the input).

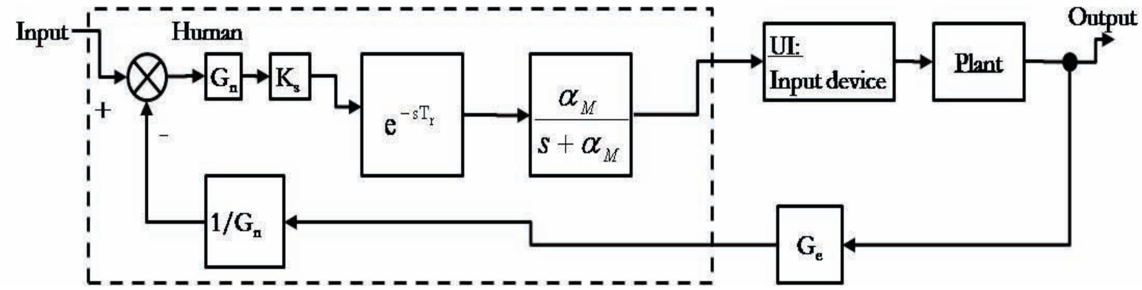


Figure 27. Block Diagram for Application Problem

The first block, the human operator informatic model (HOIM) contained in the dashed line box of Figure 27, is derived from numerous human subjects experiments. The central nervous system gain G_n is the inverse of the gain in the plant and feedback loop so that the system has closed loop unity gain. The human operator strategy transfer function K_s is a parameter to measure the effect of various pre-learned techniques that might alter user performance. The total reaction time delay T_r is determined by:

$$T_r = T_{r0} + \left(1/\beta\right) * \left(H_s\right) \tag{11}$$

where

T_{r0} is the initial human reaction time delay,

β is the choice reaction time baud rate, and

H_s is the information content of the task, measured in bits. This parameter is explained further in the later section on task description.

As it can be seen in the model, estimated decision reaction time is modeled differently than estimated movement time. Simple human movement is modeled as a low pass filter with an empirically-derived movement informatic frequency α_m. For this problem we assume a normally-performing trained and motivated operator; however, the model has been

built to accommodate specific sub-population users (i.e.: those with anomalously exceptional or inferior capabilities). The default parameters are:

- $K_s = 1$
- $T_{r0} = 0.20$ s
- $\beta = 6.7$ bits/s
- $\alpha_m = 5.0$ Hz.

For a more detailed discussion of these parameters, see (Phillips, Repperger, Kinsler, 2007 or Phillips, 2000).

In this problem, we model the user interface as a proportional-integral-derivative (PID) controller transfer function as shown in Figure 28. PID controllers are a standard form of feedback control used in industry (Franklin et al., 1994; Tan, Qing-Guo, & Chieh, 1999). It is mathematically equivalent to place this block in either the controller or interface position in the block diagram (Franklin, Powell, & Emami-Naeini, 1994). The challenge is to determine desirable PID parameters given the particular machine dynamics and the nonlinear human input response to a wide range of task complexities. In this representation, the proportional parameter influences the device speed of response (the responsiveness to operator input), the integral parameter influences the steady state error (how closely the

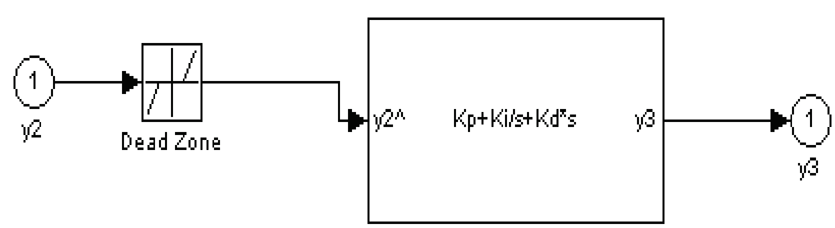


Figure 28. Block Diagram for the Input Device

output ultimately matches the goal), and the derivative parameter influences the damping of response (how much the output oscillates). Tuning is the process of balancing all three parameters to achieve the desired closed loop response.

The third block, the plant transfer function shown in Figure 29, contains parameters to describe the particular dynamics present in the machine. This model contains a parameter for plant gain K_{plant} as well as modeling of real world properties such as delays, backlash, rate limiting, and saturation.

The final block, located in the feedback loop in Figure 27, contains a parameter for environmental gain (G_e) which models the effect of the environment on operator perception of the response.

To model the wide range of task complexities, the parameters in the HOIM block are a function of the information content of the task, H_s. For a set of equiprobable responses, this is simply the $\log_2$ of the number in the set. When the possible responses are not equally

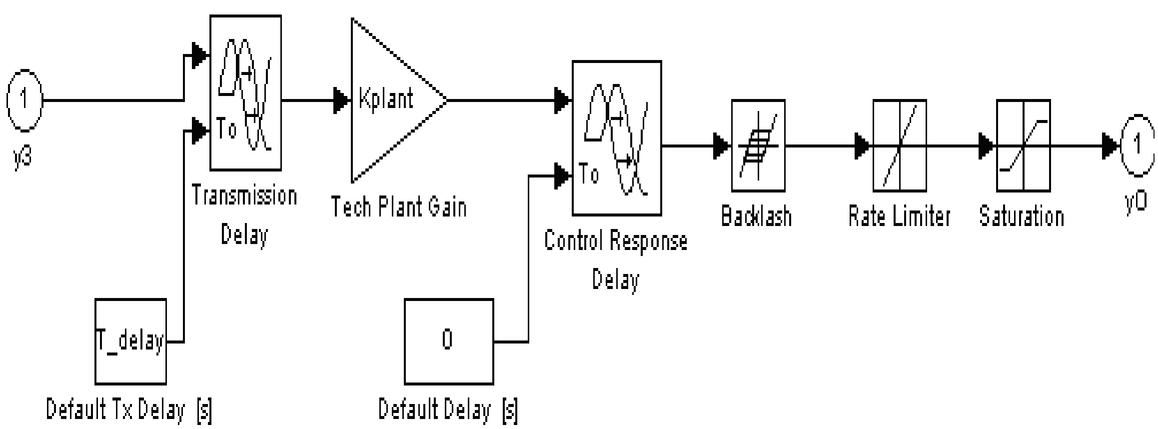


Figure 29. Block Diagram for the Machine

likely, the determination of H_s is is a weighted summation of the set. Quantifying task decision complexity in this way has been used in numerous models of the human informatics system (Phillips, 2000). This simulation finds the interface parameters that minimize the worst-case of the task performances over numerous scenarios of varying decision complexity (i.e., H_s from 2 to 32 bits). The scenarios are weighted based on the probability of occurrence so that the method predicts parameters for the best performance over the range of all probable mission flows.

The appropriate performance metric of this problem is the speed to reach a new steady state. This can be defined mathematically as the settling time of the system response. Because the model contains the nonlinearities of human and plant there is no guarantee of response proportionality or time-invariance. This means that we cannot extrapolate the response to a unit step input to other input types. We must account for this limitation by providing a full set of representative command inputs and necessary response performance in the constraint set. For this application problem, however, all task goals are assumed to require the same magnitude of response and the task is completed when the output settles to within 2% of desired.

Algorithm

Algorithm selection is based on problem type. Expressed mathematically, the requirements of this problem are to minimize the worst-case task completion time over a range of prioritized conditions. This is called a *minimax* constraint problem and can be solved by a weighted Tchevycheff method (also called the weighted min-max method) which is expressed in an optimization problem as:

$$\begin{aligned}
&\min_{x}\left(\max_{i\in S} F_i(x)\right)\\
&\textit{subject to}:\\
&c(x)\le 0\\
&A\cdot x\le b
\end{aligned}\tag{12}$$

where:

x is a vector of the interface parameters $[K_p \;\; K_i \;\; K_d]$

i enumerates the scenarios in set S

$F_i(x)$ is the "objective function" evaluated by model simulation. In this case the input is the x vector and the output is the settling time ($t_{2\%}$) of the step response.

$c(x)$ is the set of all non-linear constraints on the response of the system

A is a matrix of all linear constraints

This can be coded in Matlab® using a modified sequential quadratic programming method. Rather than provide a closed form objective function, the algorithm loads all parameters into the Simulink® model. The simulation results are then output and the time-domain response is analyzed. The algorithm then continues its operation based on the results. The key to successful execution of the proposed method is the interaction between the optimization algorithm and the model simulation. The interface between the algorithm and the simulation environment must correctly condition the inputs and outputs for correct operation.

Results

We compare the proposed method against a baseline of performance created by using a linear synthesis method to select the interface parameters. To do this we found a simplified closed form mathematical model of the overall transfer function. This is:

$$\begin{cases} G(s) = \dfrac{G_n K_s K_{plant} \alpha_M}{(s+\alpha_M)} \cdot \dfrac{-(s-\theta)}{(s+\theta)} \cdot \dfrac{K_d s^2 + K_p s + K_i}{s} \\ H(s) = \dfrac{G_e}{G_n} \\ GH(s) = \dfrac{G_e K_s K_{plant} \alpha_M}{(s+\alpha_M)} \cdot \dfrac{-(s-\theta)}{(s+\theta)} \cdot \dfrac{K_d s^2 + K_p s + K_i}{s} \\ T(s) = \dfrac{G}{1+GH} \end{cases} \tag{13}$$

where:

$\theta=1/(T_d/2)$ is the rational function approximation (the padè approximation) of the second-order equation, and

G(s), H(s), and T(s) are the forward, feedback, and closed-loop transfer functions, respectively. Using the Ziegler-Nichols heuristic for initial PID parameters, and adjusting to achieve a stable response, the interface gain matrix was determined to be: [1.2 0.002 0.002]. The step response of this result applied to the nonlinear system is shown in Figure 30. Notice that the system is stable and has a rapid response time; however, it also demonstrates a large overshoot and long settling time.

Applying the proposed method to this problem resulted in convergence at a local minima with the interface gain matrix determined to be: [0.5031 0.0043 0.0226]. The model response of this solution set is shown in Figure 31. The weighted min-max method only guarantees local (not global) minimization; however, a comparison of Figures 30 and 31 reveal that the initial tuning is significantly better than those obtained using the simplified closed form for this type of problem. The solution set determined by this method predicts a 69% better settling time than the baseline with much less overshoot and oscillation. More significantly, this method studied the responses of the entire range of task complexities.

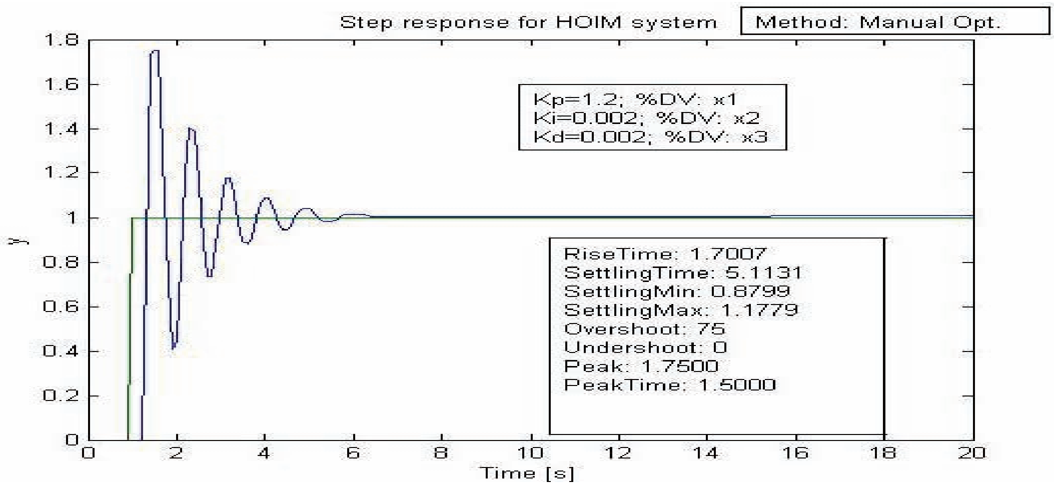


Figure 30. Step Response Linear Synthesis Initial Design

Conclusion

As it has been shown, the method presented here makes several contributions to input device design. The method is a guide to structuring the system so that the input device performance can be quantitatively studied. Use of the method incorporates human components directly into the system under study and allows for nonlinearities and empirical

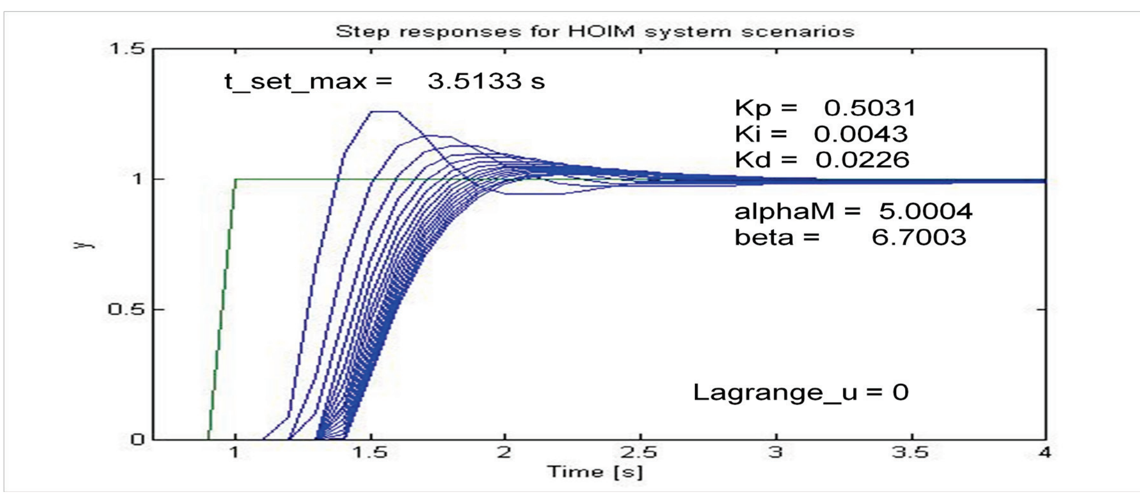


Figure 31. Step Response of System Designed by New Method, Initial Design

data. Finally, the method provides an economical way to design for improved human-machine interaction early in system design.

VII. Method 4: User Interface Design: Displays[8]

In this chapter, the methodology presented in Chapter 3 is applied to the layout of information and functions in multi-function displays. It is shown that use of the proposed method improves predicted system performance.

Introduction

User interface design is a critical component of modern system development. It has been identified as a key factor in both system safety and efficiency (Shneiderman & Plaisant, 2005). These findings have fueled the interest in better ways to design UIs (also called HMI, human-machine interfaces) and the recognition that human-computer interaction (HCI) is a significant design consideration (Saunders, 2005).

In (Hardman et al., 2008a) we presented an approach to display layout design based on human performance modeling. The paper proposed metrics, such as the HCI Index, that allowed for a quantitative evaluation of layout effectiveness. In (Hardman et al., 2009a) we focused on how to use those same metrics to generate optimal layout predictions early in system design. We did this using a hybrid algorithm (consisting of a seeded genetic algorithm in conjunction with a pattern search algorithm) that predicts new display layouts with minimized control operation times. In this paper we build on those earlier works to present a complete method. We then compare this analytical method with human subjects experiments through a meta-analysis of aircraft display redesign projects.

---

[8] The contents of this chapter have been independently submitted for publication by Hardman, Colombi, Jacques, Hill, and Miller .

Background

User interface design has always required difficult tradeoffs. In aircraft design, interface controls were traditionally switches and buttons and interface displays consisted of numerous lights and gauges. These had to be prioritized and placed in the user's limited field of view according to their rated importance. The final design was a tradeoff allocation of the limited visual "real estate" (DoD, 1998). The trend in modern avionics has been to reduce the clutter by consolidating controls and displays on multi-function displays (MFD). We see an example of this evolution in Figure 32. The photo on the left shows an example of fully dedicated controls (gauges and switches) and displays in the A-10. Note the two rows of lights for the Armament Control Panel (ACP). The center photo shows how designers of the next generation aircraft consolidated the ACP and additional functionality on an MFD with bezel buttons around the edges for the F-15. These are called "soft buttons" as their functions can be tailored to the particular page being displayed. Finally, the photo on the right is the proposed next generation cockpit in the F-35. Note that most dedicated gauges and switches are replaced by two large touchscreens. Designers have chosen to make the MFD larger and added touchscreen capability (Lockheed Martin, 2006). This design eliminates the need for peripheral buttons because the user can directly manipulate options on the page. This yields a more intuitive interface because it emulates real life actions (Shneiderman & Plaisant, 2005).

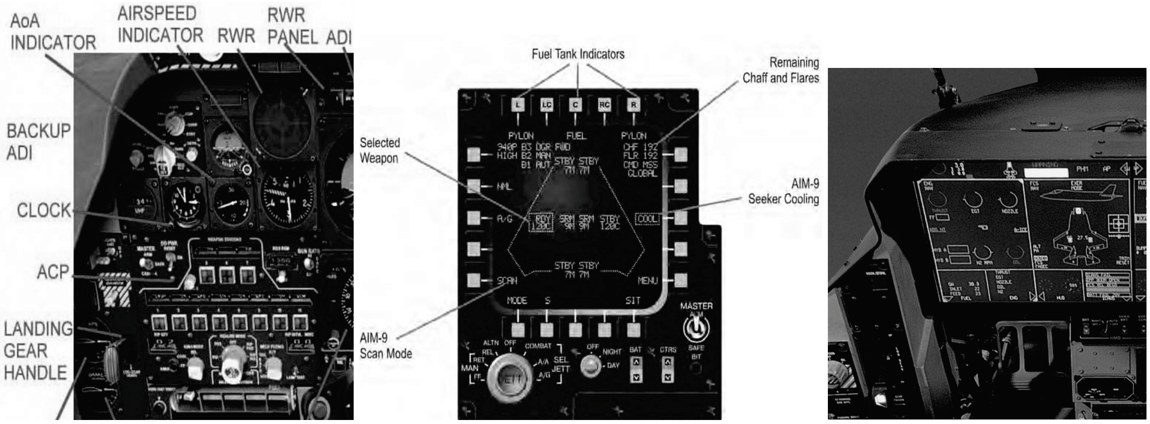


Figure 32. The Evolution of Cockpit Design
(Left and center: USAF photos. Right: Lockheed Martin photo, used with permission)

The increased computational sophistication and prevalence of such technologies give engineers a design freedom never before experienced. However, like any frontier, the lack of constraints increases the decision complexity and highlights the need for sound principles. The F-35 cockpit is an example of a "glass cockpit" where information and functions are combined and superimposed in novel ways. Touchscreen technology is now prevalent in other forms of transportation (such as automobile dashboards and boat navigation/radar systems) and public kiosks such as automatic teller machines and airport check in stations. Whether by intention or default, new HCI technologies have changed the user interaction from simple selection to time-division single access. Intelligent information grouping can improve decision making, but poor layouts can be a restriction on operations. The task of the designer is to find a layout that best exploits the informatic relationships of the intended operational activity threads. In the aviation domain, experimental psychologists have done this work by correlating necessary information by phase of flight (Schvaneveldt, Beringer, & Lamonica, 2001). Designers can use the context-aware information, combined with human performance modeling, in cockpit design.

UI design is not a mature science; it is still performed mostly by intuition in the lab and trial-and-error in the field. This is an expensive, iterative process. Though actual user testing is regarded as the gold standard of evaluation methods, the time and expense of user testing make it prohibitively difficult to be used exclusively. Often, usability testing is delayed until late in the development process. Practitioners generally use qualitative inspection evaluation methods, such as expert surveys or cognitive walkthroughs, in early system development (Bisantz & Burns, 2009). Researchers have produced UI design guidelines that are drawn from empirical studies. These guidelines are intended to help practitioners better examine the layout, grouping, and ordering of the UI (Francis, 1999; Shneiderman & Plaisant, 2005). A summary of the applicable layout guidelines include:

- Memory load on the user should be reduced,
- Input actions by the user should be reduced,
- Functions used together should be grouped together,
- Number of UI levels should be reduced, and
- Some items should be given dedicated consideration due to frequency or time criticality.

Additional UI guidelines advise on the ordering of buttons.

- Place buttons to increase the probability of repeatedly selecting the same button
- Related functions should be next to each other,
- Ordering should be consistent and standardized,
- Make consideration for likely user errors, and
- Provide feedback and closure.

These guidelines provide general insight, but they do not help the designer map information and functions to specific operational threads (Kieras, 2004). Furthermore, as a whole they often yield contradicting advice with no established structure for prioritization.

Studies have established the connection between UI design and both system performance and error rates (Sirevaag et al., 1993). The UI design community is very interested in being able to evaluate this more quantitatively (Shneiderman & Plaisant, 2005). Our proposed metrics quantify the established guidelines on menu structure so they can be used to objectively evaluate and predict optimum display designs. The guidelines related to button order and style can then be used in specific page design. Modeling the layout allows both improved insight and quantitative analysis using numerous commercially available software packages.

Mathematical Formulation

Manipulating specifications into design is a complex problem. For example, a basic aircraft MFD will have over 60 possible pages (Hardman et al., 2008a). This problem has more than $10^{1,000}$ feasible designs. The U.S. Army's AH-64D Apache helicopter has over 300 possible pages (Francis, 1999). This quickly makes manual or exhaustive search methods impractical.

Model Construction

The choice selection process in a display layout can be considered as a quasi-stochastic process in which, given a present state, the future states can be reasonably assumed to be independent of the past. This is called a discrete Markov decision process or a Markov chain. Specifically, the problem as we have defined it is an irreducible, non-absorbing, time homogenous Markov chain (Puterman, 1994). A Markov chain is usually modeled as a state diagram with *nodes* (also called vertices) v_x and lines between node pairs called *edges* (also

called transitions) $e_{a,b} = [v_a\ v_b]$. In the following definitions, we model the interface from the system perspective, so we call user actions "inputs" and data displays "outputs."

<u>Nodes</u>

Each interface output is a combination of three components: information, menu options, and functions. We will refer to each unique interface output as a *page* of the display when the context is clear as it is consistent with website terminology and makes the discussion more readable. Although the output of the user interface is predominately visual, a page includes the total presentation to include audible, haptic, and tactile modalities. Display information is processed data that is useful for decision-making. This is agnostic to the ways that the information can be presented. For example, on various pages, altitude may be presented as a number, a scrolling tape or a voice annunciator. The other two components of a page are choices for the user. Menu options are the available transitions to other pages. Functions cause the execution of some action, but the interface remains at the same page. We model each page as a separate node in the model. The complete node set, then, is the set of all pages defined in the system specification.

<u>Edges</u>

Interface inputs are the selection of menu options by the user. This includes the activation of switches, buttons, soft buttons, selectable icons, and even voice recognition commands. Interface inputs are distinct from task-related inputs, such as vehicle steering or target designation, but in this paper, we will use *input* when the context is clear. These inputs are modeled as edges; the transitions from one node to another. The complete edge set, then, is the set of all possible direct transitions from all defined pages in the system.

Graph Description

If we can translate the system specifications correctly, more insight can be gained from using the mathematics of graph theory. Mathematicians would call the model of our problem a directed graph or *digraph*. By design, it will be a connected digraph as all nodes must be accessible. In general, models will not be symmetric nor simple graphs; both loops and multiple edges can exist (West, 2001). Self-loops represent functions, and multiple edges define a scenario in which more than one input causes the same page transition (e.g., operator can use a voice command, shortcut key, or a menu button). We use the term *HCI-defined G* to refer to a graph *G* that has been built from the definitions stated here and is thus a representative model of a valid interface design layout. In Figure 33 we show an HCI-defined graph of an F-15 MFD layout. For this problem, note that the task analysis indicated that the normalized probability of transitioning between nodes was approximately zero for a large number of the page combinations. Similar mappings of interfaces were used

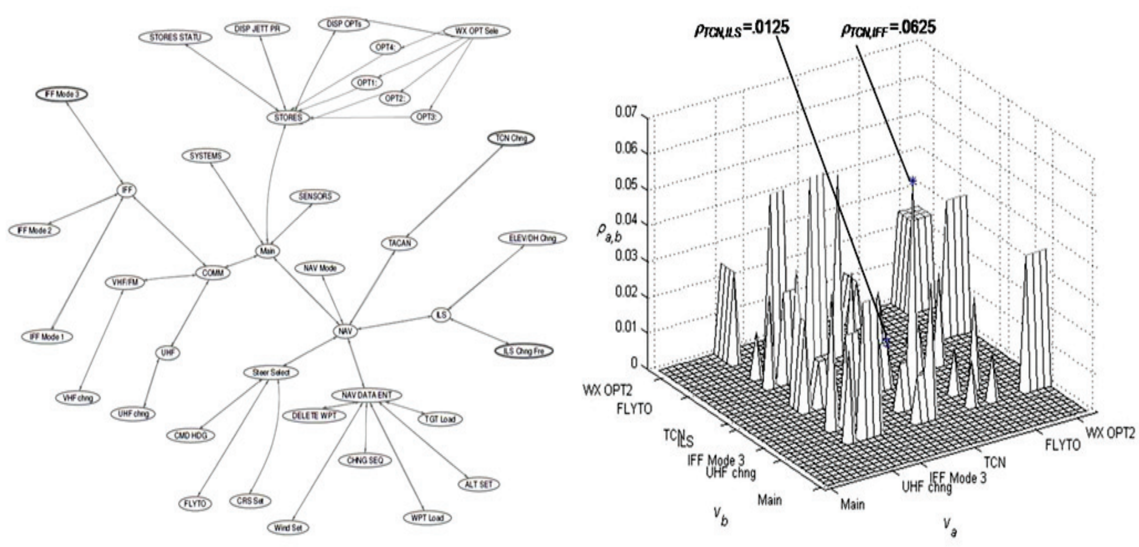


Figure 33. Graph of Menu Layout (left) and Plot of Affinity Matrix (right)

by mathematicians to model internet webpage hyperlinks and study their growth patterns (Albert, Jeong, & Barabasi, 1999).

Mathematically, a graph can be expressed in several matrix forms. The most intuitive is an *adjacency matrix* of binary numbers $A(G)$. For each i,j element of the matrix equal to one, there exists an edge between v_i and v_j (West, 2001). Values on the diagonal represent the number of functions on a page. As self-loops, these do not invoke a page transition. A primary design criterion of HCI is to make the displays intuitive. This will lead to semi-disjoint sets due to the logical grouping of related pages.

Each edge has a probability of transition $\varrho_{a,b}$ that is the likelihood of accessing node v_a from node v_b. For example, in Figure 33, $\varrho_{\mathrm{TCN,ILS}}=0.0125 < \varrho_{\mathrm{TCN,IFF}}=0.0625$ which quantifies the fact that it is operationally less likely for the F-15 pilot to transition directly to a precision approach (ILS) after a navigation update (TCN) than it to change the transponder frequency (IFF). A matrix of these probabilities is called the transition probability matrix, or simply *transition matrix*. For a layout of n pages, the transition matrix is an n x n matrix of $\varrho_{a,b}$ values. For existing systems, the transition matrix can be formed empirically by recording actual work flows. For predictive purposes the transition matrix can be formed as follows: first, decompose the system's intended purpose into operational threads or action sequence alternatives. Human factors and cognitive systems engineers have produced a large amount of research on how to best do this. The recommended methods include task analysis or cognitive task analysis (Stanton et al., 2005). Secondly, total the count of each specific page change during the execution of all identified tasks. Divide each element of the matrix by the sum of its row. Now the entry of row a column b is the expected probability of transitioning

from *a* to *b*. That is, P$(a|b)$. Transition matrices may also be formed by gathering these probabilities during simulations involving prototypes.

The transition matrix gives insight into the states that we are most likely to reside in, but it does not capture the most likely transitions. To get that information, we use a special form of the transition matrix, called the *affinity matrix* and represented by $\boldsymbol{P}$ (capital rho). To obtain $\boldsymbol{P}$, we tally all transitions during operational threads as described above. We then normalize the joint probability mass function of the tallied matrix. In this way, the sum of all entries in $\boldsymbol{P}$ will equal 1.0 and each entry is the weighted probability of the transition occurring in the intended use. For the F-15 MFD previously mentioned, a surface plot of the resultant affinity matrix is shown in Figure 33.

Method

Once the proposed layout has been defined as above we are able to find potential designs. The most valuable contribution of model-based evaluation is the ability to quantitatively score proposed layouts, even before initial prototyping. In graph theory, path length is defined as the number of edges on a path between two nodes, and distance is defined as the length of the shortest path. For an interface layout this corresponds to the number of inputs required to move from one page to another. A sum of the shortest path distances between all nodes in a graph, now called the Wiener Index, is a simple goodness estimate for Markov chain analysis (Puterman, 1994). Liu proposed that a similar type of scoring should be used to evaluate the search efficiency of menu layouts (Liu, Francis, & Salvendy, 2002).

In (Hardman et al., 2008a) we proposed a novel metric to measure proposed layouts which we termed the HCI Index. This index is the expected mean menu selection time of a layout. To obtain this we use a weighted shortest path calculation based on the Floyd and Warshall algorithm, as coded by Iglin (2007). The paths are weighted by mean system processing speed and expected human choice reaction time. System processing delay times are obtained from network analysis, but they are usually only significant in teleoperation scenarios (e.g. geographically-separated control of unmanned vehicles). Human choice reaction times are the result of perception, decision making, and activation (by movement or articulation) (Phillips, 2000). The work of many researchers, such as, Hick, Hyman, Deininger, and Wickens, provides us with the components necessary to make accurate estimates of the human performance.

Menu Selection Time

Researchers have independently established the logarithmic relationship between the number of elements in the decision list *n* and the time it takes to make a selection, the choice reaction time, RT_c (Seow, 2005). This relationship is accepted by many as the Hick-Hyman Law. It has been supported by extensive subsequent human subjects testing and published in many forms. The instantiation most applicable to our problem is:

$$RT_c = a + b \cdot \log_2(n+1) \tag{14}$$

where:

a is an empirically derived constant for the task-specific simple reaction time,

b is an empirically derived constant for the task-specific human information processing capability, which is often expressed as 1/β where β is the human baud rate (Card, Moran, & Newell, 1983).

This relation assumes a well-trained, but non-expert, user and menu items that are ordered with some underlying logic. Our analysis uses mean population parameters (on average) of 0.212 s for simple reaction time and 6.7 bits/s for human baud rate (Phillips, 2000). This permits the following definition: The *weighted distance* from v_0 to v_k on an HCI-defined G, is written as $d_w(v_0,v_k)$, and defined as:

$$d_w(v_0,v_k)=\sum_{i=1}^{k}\left(t_i+\left(0.212+(0.153)\log_2\left(d^+(v_{i-1})+1\right)\right)\right) \tag{15}$$

where:

v_0 and v_k are any two arbitrary nodes,

t_i is the system processing time delay constant associated with the i^{th} edge on the (v_0,v_k) minimum path, and

$d^+(v_{i-1})$ is the number of selectable options of the tail node of the i^{th} edge on the (v_0,v_k) minimum path.

This metric is written as $d_{w\_G}(v_0,v_k)$ when the associated graph G is not clear from the context. A weighted distance calculation using (14) can be used to derive a metric for evaluating the entire layout with respect to task transition time. We write this as:

$$D_w(G)=\frac{1}{n^2}\sum_{v_a,v_b\in V(G)} d_w(v_a,v_b) \tag{16}$$

This index is the average time to move between any two pages. While useful for interface design, minimizing (16) does not guarantee that a layout is optimized for its mission, or even any specific task. For that, the metric must include input from its expected operational context. Designers can provide this in the form of the affinity matrix described above. This weights the transitions according to those most likely to be accessed in the

course of executing the system's purpose. In this way, we have quantified the guidelines for menu layout discussed previously. We named this the **HCI Index** and define it as:

$$D_{w,\rho}(G) = \sum_{v_a, v_b \in V(G)} d_w(v_a, v_b) \cdot \rho_{a,b} \tag{17}$$

In (Hardman et al., 2008a), which is included in Appendix D, we proved that, given a set of proposed layouts, the one with the smallest value of (17) is guaranteed to have the smallest expected control operation time.

Data Entry Load

For most UI designs, we must account for functions that require more from the user than simple selection. Many volumes of data, such as (Boff, Kaufman, & Thomas, 1986), have been compiled covering the many ways that data may be entered into a machine. This is added to (16) to get a layout's expected control operation time for the display as a dedicated task.

Secondary Task Factor

For tasks in which the control operations are not the primary task, such as while flying or driving, we must also account for the delays due to divided attention. We use Wickens' multiple resource theory to categorize the multi-tasking interaction based on input, processing, and required output (Wickens, 2002). Existent human subjects data can then be used to predict a secondary task factor for the particular problem being examined. This factor is a multiple of the expected control operation time.

The final result, after the calculations of the previous three sections, is the expected control operation time for display usage as a secondary task. In the next section, we will

demonstrate this process and compare the expected values with corresponding human performance.

Human Subjects Comparison

In order to gain confidence in our method, we sought to correlate it with actual human subjects testing. We were able to do this by reusing a series of studies performed at the Air Force Wright Aeronautical Laboratory. These studies were part of the cockpit upgrade efforts for the A-7 and F-15 fighter aircraft (Calhoun, Herron, Reising, & Bateman, 1980; Herron, 1978; Murray & Reising, 1980; Reising & Curry, 1987). They primarily focused on the merits of two competing control logics for MFD layout. To study the operational effectiveness, human subjects (pilots) were given operationally realistic tasks to perform in high fidelity simulators such as the one shown in Figure 34. The first layout, designed by what they called *branching control logic,* was a traditional hierarchical menu tree structure. In this layout, specific information and functions are accessed by paging through the logic tree of pages grouped by category. The second layout, designed by so called *tailored control logic*, grouped information and functions by those commonly used together given a particular phase of flight. If an unavailable function was needed, it could still be accessed by the branching control logic hierarchy.

As would be expected based on the UI design guidelines, layouts based on tailored switching logic exhibited a significant speed advantage, as measured by mean time required to perform a given control operation. The F-test and p-value results for the F-15 were $F(1,49)=126.678$, $p<0.01$. On average, it took almost twice as long to perform the same

Figure 34. Simulator Used for F-15 Cockpit Development
(US Air Force photo)

tasks using the layout based on branching control logic in the A-7 (24.14 s versus 16.53 s) and almost three times as long in the F-15 (30.63 s versus 10.86 s).

The cost to perform each of the display experiments, using human subjects tests, was more than $400k (1987 $) and five months of effort (Reising, Calhoun, & Curry, 2009). Using the same specifications as those in the studies, we modeled the design problem using our proposed approach. We calculated the expected control operation times using the HCI Index, the estimated data entry load, and the secondary task factor as shown in the left graph of Figure 35. For our cockpit design problem, we modeled the data entry load by applying the work of Deininger (Deininger, 1960). The secondary task factor was estimated using data from studies performed by Horrey and Wickens (Horrey & Wickens, 2004). Our proposed method compares closely with the actual data and predicts the improvement in performance obtained with the tailored control logic. This is summarized in the right graph of Figure 35.

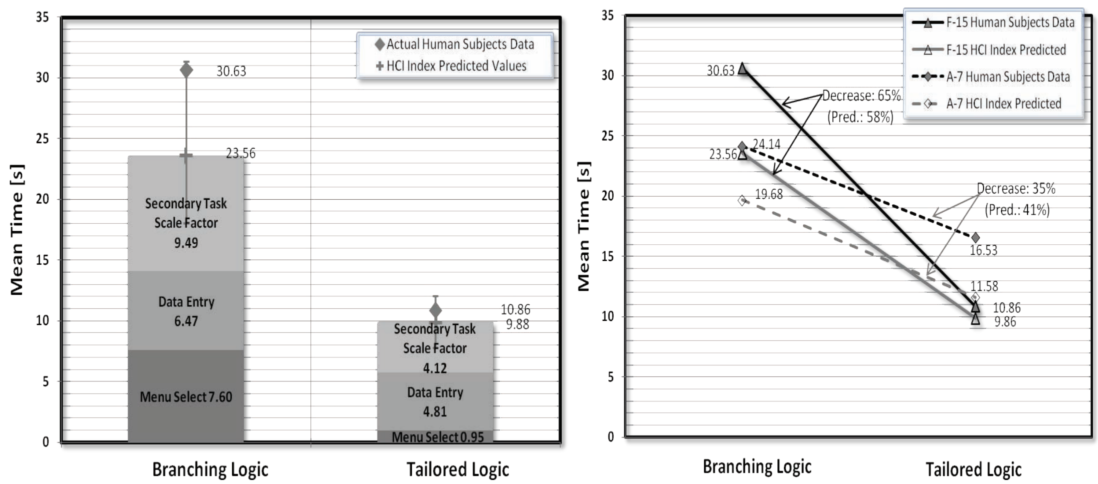


Figure 35. Control Operation Time (left) and Predicted Value Comparison (right) (Error bars indicate the predicted range around the expected value based on human and task variability.)

The original studies also found correlation between display layout and user error rate; and indicated a relationship between display layout and aircraft loses. The number of errant control operations performed by the subjects was much lower for the layout based on tailored logic. The layout used was found to be significant in both studies ($F(1,45)=27.412$, $p<0.01$ for the F-15). Also, the pilots were much less likely to be shot down while performing control operations when using the layout based on tailored logic. We reproduce the results of the F-15 error data in Figure 36.

Optimal Design Problem Formulation

In (Hardman et al., 2009a) we defined the challenge of UI design as an optimization problem: to find the *HCI-defined* graph *G* which minimizes the HCI Index. Our formulation is unique in that no a priori graph structure is assumed and the optimization acts on the set of edges. Designers must first have determined a proper node set of a graph *G*, an

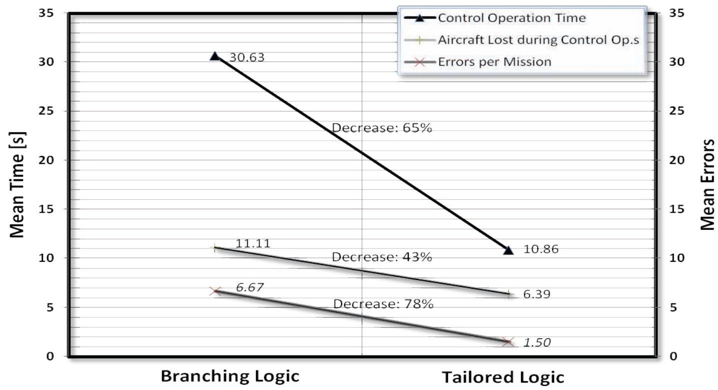


Figure 36. Comparison of Control Operation Time, Errors, and Aircraft Lost

operationally-relevant affinity matrix ***P***, and a list of time constraints on critical pages. Then, the following formulation expresses the optimal design problem.

Design Variables

Design variables are the things that can be controlled. In a menu design problem, as we have defined it, these are the edges $e_{a,b} = [v_a\ v_b]$ for all a and b.

Objective Function

The objective function is the mean control operation time. This is predicted by (16), the HCI Index. Since the HCI Index is an implicit function, it cannot be explicitly expressed in terms of the design variables. We adapt it for use in optimization routines by coding it as a software routine which receives the design variables as inputs and returns the HCI Index value.

Constraints

All real-world UI design solutions are subject to constraints. The solution must be a connected graph (no isolated components) with no stubs (a node with no outward edges).

Also, any real world problem will have an upper bound on the number of programmed transitions from any one page. The solution may also have constraints based on the required accessibility of critical functions. These can be accommodated in our model by assigning limits to the respective node maximum distance. In graph theory, the *eccentricity* of a node v_a is the maximum of the distances between v_a and all other nodes of the graph. Using (12), the *weighted in-eccentricity* is the maximum $d_w(v_b,v_a)$ for any v_b of G:

$$ecc_w^-(v_a) = \max_{v_b \in V(G)} \{d_w(v_b, v_a)\} \tag{18}$$

Simply stated, this is a measure of the greatest separation, in time, that exists between a particular page and any part of the rest of the graph. The weighted in-eccentricity of critical nodes v_{crit} must be no greater than some maximum access time b_i [s]. Thus the objective function (14) is constrained by:

$$s.t. \quad ecc_w^-((v_{crit})_i) \le b_i \quad i:1 \text{ to } m$$

where m is the number of constrained nodes.

Since this is what developers truly want to control, this is a quantitative way to implement the guideline on critical functions.

Problem Characterization

Existence of a solution

For the unconstrained form of the previously defined problem, the existence of an optimal solution is guaranteed. To see this, assume an HCI-defined graph that is maximally connected (i.e., every node is adjacent to every other node in the graph). This design, though functionally poor, is a feasible solution.

For the constrained form, an optimal solution exists as long as no constraint precludes the existence of a feasible solution. This minimum upper bound on transition for critical nodes b_{i_min} [s] is a function of the number of critical nodes n_{crit} in the problem:

$$b_{i\_\min} = t_i + (0.212 + (0.153)\log_2(n_{crit})) \tag{19}$$

Therefore, the limitation on the existence of a solution is that no constraint b_i on any of the critical nodes n_{crit} can be less than b_{i_min}. Put simply, there is a limit to how many places can be given priority access, regardless of the solution.

As mentioned, any real world solution will be constrained by the limitation on the number of transitions from a particular page. This puts an additional limitation on the number of critical nodes that can be constrained at b_{i_min}.

Uniqueness of a solution

An optimal solution on a weighted graph is guaranteed to be unique if and only if all edge weights of the graph are unique. This is established by the fact that every underlying spanning tree represents a unique run of Kruskal's algorithm (Kocay & Kreher, 2000). If all edge weights are different there is only one possible unique result from Kruskal's algorithm; ergo, the solution is unique. In an HCI-defined graph, the weighted "distance" is $d_{w\_G}(v_0,v_k)$ as defined in (14). Since there is no uniqueness limitation on (14), there is no guarantee on solution uniqueness.

Solution Space

The number of design variables is the number of potential node pairs or n^2. This results in a total set of 2^{n^2} possible graph. Because feasible solutions must be at least weakly connected, many solutions are eliminated from the feasible set. However, even modest

problems have a very large feasible set of solutions. In fact, computational complexity theorists expect that, for this class of problem, no polynomial-time solution algorithms are possible (Garey & Johnson, 1979).

Search for the Proper Tool

In addition to being complex, both the objective function and the constraint functions are nonlinear. That is, neither the property of additivity nor homogeneity are satisfied. Also, the design variables are discrete. This precludes the use of traditional optimality methods and even numerical search methods (e.g., integer programming and sequential linearization methods) which require gradients or gradient estimations (Michalewicz & Fogel, 1998).

This type of problem is addressed by heuristic methods such as simulated annealing, genetic algorithms, or tabu search. We evaluated these optimization techniques and determined that this problem is best accommodated by the attributes of the genetic algorithm (GA). A GA is a meta-heuristic based on concepts of genetic theory. An entire population of potential solutions is characterized by design variables in the same way that chromosomes characterize an organism. The GA allows us to perform global parallel searches in a highly discontinuous solution space and resist premature convergence to a local minimum (Payne & Eppstein, 2005). Also, we are able to tune the initialization and selection criteria to exploit a priori knowledge of partial solution clustering. The work by Matsui and Yamada confirmed the superior performance of a properly tuned GA for menu design problems (Matsui & Yamada, 2008).

Algorithm Tuning

As explained in (Hardman, et al., 2009a), parameter tuning is a very important aspect of almost all heuristic approaches to problem solving. One strength of GA is its versatility; however, the cost of versatility is the challenge of determining the correct parameter settings for a specific problem. The optimization was performed using the Matlab® optimization toolbox and the graph theory toolbox (Iglin, 2007). The graphics were generated by the Matlab® Biograph utilities.

Design Variable Representation

A proper representation maps the state space of the variables used in the GA population to the solution space of the actual problem. Michalewicz and Fogel have identified key qualities of a good representation for GA problems. They state that a representation should:

- Be a bijective mapping (one-to-one and onto) to the real world variable set,
- Maintain the link of parents to offspring; usually through a graduated range,
- Be segment-able in that components of the whole are independently useful, and
- Always, or at least frequently, produce feasible solutions (Michalewicz & Fogel, 1998).

These are important for the computational efficiency of the GA. The most intuitive representation in this problem is a nxn length string of binary numbers representing the elements of an adjacency matrix; however, there are numerous other graphs representations in use such as a node list, an edge matrix, and a star representation array (Kocay & Kreher, 2000). All were examined for sufficiency. We discovered that none of the traditional mathematical forms provide the desired qualities for a GA population.

In the pursuit of a suitable chromosomal form, we discovered the best performance by representing the design variables of the display layout by an n-length vector of real numbers between one and the edge count constraint defined in the specifications. Each value in the sequence represents the number of edges originating at the corresponding node. The corresponding edge set, then, is determined by using the affinity matrix as a look up table. For each node, the corresponding row of the affinity matrix is rank ordered by transition probability. Edges are then assigned up to the stated number. This is not a natural representation, but it is superior to all traditional mathematical forms in that it possesses all of the desired qualities of a GA population. Furthermore, it uses contextual knowledge to guide the connectivity of the graph toward likely quality designs.

This representation does depend on an accurate task analysis to produce quality transition probabilities. To make this representation less sensitive to measurement noise, an additional variable is added to the representation to make it an ($n+1$)-length vector. This additional term sets the level of random noise introduced to the rank ordering of the affinity matrix elements. Thus, the GA is populated with designs from across the entire range of feasible solutions.

Since this is a custom population type, we had to modify the Matlab® creation function, mutation function, and crossover function. In addition, we had to create transformational routines that convert between the vector representation and the adjacency matrix form used by the shortest path algorithm.

Initial Population Creation

Heuristic methods are not very computationally efficient in terms of feasible solution convergence time. As studied by (Payne & Eppstein, 2005), the genetic algorithm will

perform better if "seeded" by feasible solutions with wide variation. To achieve this, we initially provided the GA with graphs generated using Prim's algorithm. The algorithm guarantees minimum spanning trees (Kocay & Kreher, 2000). We found that this method did not produce an initial population with enough variation to prevent premature convergence to a local minimum. We instead provide the genetic algorithm with an initial population of pseudo-randomly generated spanning trees.

In addition, to further counteract premature convergence, two sub-populations are simultaneously evolved with full independence except at periodic migration points. Every 20 generations a mutual migration of 10% of the population occurs. These steps have successfully made the algorithm more resilient to the trap of sub-optimal local minima.

Fitness Function

The fitness function determines the solution space landscape. A good fitness function should yield some sort of correlation to the quality of the solution, that is, a score should be proportional to the distance from the optimal solution. The objective function described in (4) was found to do this without modification.

$$D_{w.\rho}(G) = \sum_{v_a, v_b \in V(G)} d_w(v_a, v_b) \cdot \rho_{a,b} \tag{20}$$

Constraint Handling

A primary difficulty of implementing a GA is constraint handling. Our problem requires several added procedures to account for all of the constraints discussed above. First, all variables are controlled to be between 1.0 and the edge count constraint. This means that every solution will yield at least a connected graph. Secondly, critical node constraints are evaluated as part of the scoring algorithm. Infeasible solutions are addressed

by allowing an adaptive penalty function where the size of the penalty is a function of how frequently the algorithm ventures into the infeasible space. This allows transitions through infeasible space to find other optima; especially early in the search time. The penalty grows linearly as the number of infeasible solutions increase in order to force the population back into the feasible domain.

Selection Procedure

The selection procedure determines how a member's fitness function score is used to determine both its survival and chance to spawn future generations. We use the roulette technique. This is analogous to assigning members an area on a roulette wheel that is proportional to their fitness score. This works without scaling because our fitness function algorithm returns a bounded score for all feasible solutions. In addition, we implemented an *elitist operator* in the selection process. We set this at 0.1 so the highest 10% of both populations are automatically carried to the next generation. This ensures that the best fitness level is monotonically increasing with each generation.

Variation Operators

In addition to migration and spawning, other variation operators are used to evolve the populations. We will use both mutation and crossover.

Crossover is a type of multi-parent breeding. Children are reproduced by a recombination of parts from fit members. We implement it using a heuristic function that generates children nearest the parent with the highest fitness function, but in a direction opposite from the parent with the lowest fitness function. We also use an elitist operator that keeps all offspring that exceed the fitness of their parents, but keeps only 1/n of the offspring that do not exceed their parental fitness.

Mutation is the process of making limited random changes to fit members. We implement mutation using an adaptive feasible function. This means that the changes are made with a weighting of the success of those changes in previous generations. This makes the mutation rate self-adaptive.

Local Search

Lastly, we implemented a two-phase hybrid algorithm in that, once the GA reached a termination criterion, the results are passed to a local search algorithm. The genetic algorithm was allowed to run for n*120 generations or until there has been a < 0.1% improvement for 0.1(n*120) generations. These values were determined by observation of genetic algorithms on similar problems (Michalewicz & Fogel, 1998). We then hand off the results to a pattern search algorithm. We chose this algorithm because it is able to search a local "topography" without the need for a derivative or Hessian. With this approach we are able to guarantee at least local optimality. This combines the explorative strength of the GA with the exploitative power of various local search algorithms.

Performance of the Design Method

In (Hardman et al., 2009a) we detailed the evaluation of our seeded hybrid GA. We demonstrated the ability to find globally optimal solutions for problems of designs under five pages; even in the presence of noisy transition matrices. We then compared our approach against the 63-page MFD example shown in Figure 38. The seeded hybrid GA found a design that achieved an 18% better performance prediction than the best result that was found using the predefined structure detailed in (Hardman et al., 2008a).

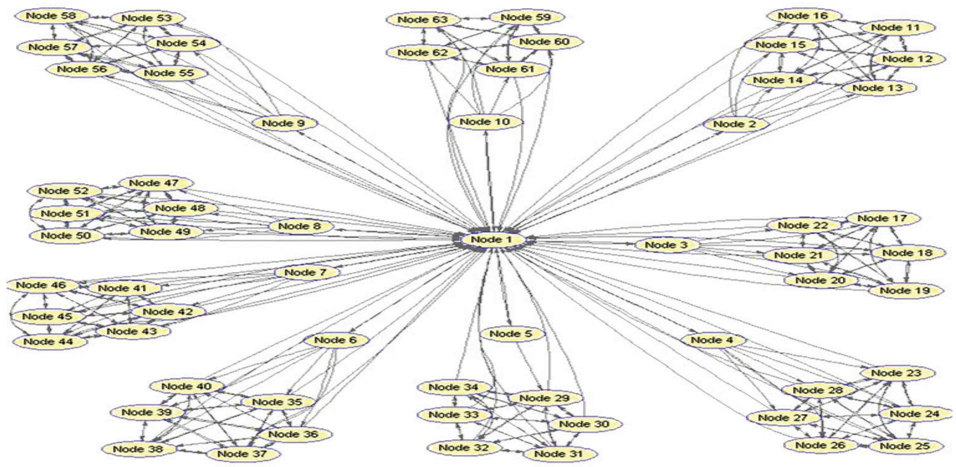


Figure 38. Genetic Algorithm Layout for 63-Page Multi-Function Display Problem

Finally, we applied our approach to the specification for the A-7 and F-15 redesign efforts and compared it against the layouts created by the cockpit design teams. The results are shown in Figure 37. As it can be seen, our HCI Index accurately predicted both the results of the human subjects tests and together with the hybrid GA method, found even better possible display layouts for both aircraft.

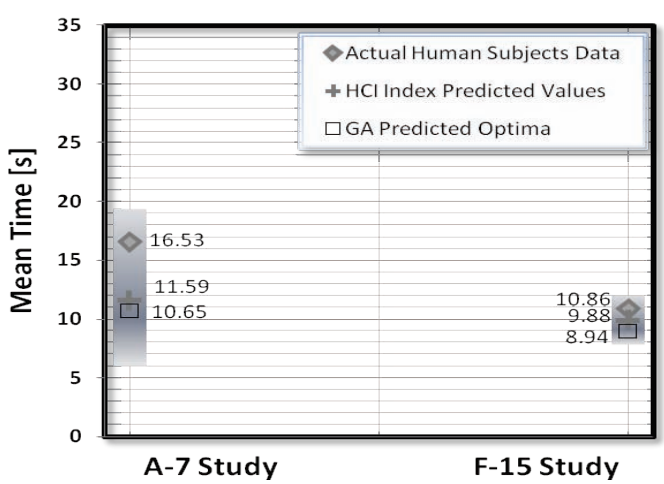


Figure 37. Comparison of Actual, Predicted, and Predicted Optimal Performance

Conclusion

As it has been shown, our method is an improved approach to UI design. Modeling the problem in this way enables quantitative evaluation that can be done faster, cheaper, and sooner in development process than qualitative methods. Furthermore, as demonstrated, we can predict optimum designs. This has the potential to improve the development many modern systems such as of control panels and public kiosks.

VIII. Conclusion

Not everything that can be counted counts, and not everything that counts can be counted.
–Albert Einstein

Conclusions of Research

This research began with a study of system development efforts and challenges. That study revealed that systems engineers need to improve the integration of human considerations during system development, as well as they need new tools and methods to perform that integration. The development of methods and tools was then accomplished that could better achieve this aim; all within a new methodology that coherently engineers HSI through the application of empirical data. These methods especially support quantitative integration early during system design.

The requirements elicitation method prioritizes requirements related to the human components of the system early in system development. It enabled a thorough review of all past mishap data in the context of a system under study. By applying the similarity weighting and mapping mishap data to HSI domains, it bridged the work of the safety community with the systems engineering processes. With updates to the human error taxonomies and mishap data, this method can remain relevant for the complex human-machine interaction of the aerospace industry.

The second method enables engineers to make decisions regarding system automation. The results for the method, as applied to collision avoidance in a UAS, show how automation analysis can be improved. In this particular application, the available technology makes the selection difficult. The method made the choice clear and objective, and as these

technologies mature, the analysis is easily updated to examine if the conclusions remain valid. Results from this dissertation will quantitatively support the debate for safe UAS flights through controlled airspace.

The third method made use of a simulation-based approach to accommodate empirical data for human capabilities and limitations. The method is a guide to structuring the system so that the input device performance can be quantitatively studied. Results were demonstrated to set requirements for user interfaces across a variety of operational tasks. It reveals an economical way to design for improved human-machine interaction early in system design.

The final method guides the layout of information in multi-function displays (MFD). It was combined with Markov chains and hybrid seeded genetic algorithms to find mathematically best display layouts faster, cheaper, and sooner in the development process than qualitative methods. Algorithm results were confirmed with human subjects test data for F-15 and A-7 avionics. This has the potential to improve the development many modern systems such as of control panels and public kiosks.

Research Contributions

This research and use of empirical human performance data improves the systems engineering body of knowledge. The initial literature review and analysis form a current picture of SE issues, and the evaluation of mishap data is original in its SE perspective. Together, these give the SE community current and substantiated evidence of the importance of HSI. The given methods equip systems engineers to address the identified issues as part of sound system engineering practice, and the applications demonstrate how to engineer HSI starting from the earliest stages of system design.

The results and conclusions of this research have led to a dozen publications that address the pertinent issues of audiences in a diverse list of venues:

1. The IEEE International Conference on Distributed Human-Machine Interfaces.
2. The INCOSE INSIGHT periodical; two articles.
3. The IIE International Engineering Research Conference.
4. The International Conference on Systems Engineering Research.
5. The INCOSE International Symposium.
6. The IEEE Systems, Man, and Cybernetics (SMC) International Conference.
7. The IEEE SMC Journal of Systems and Man.
8. The AIAA Journal of Aerospace Computing, Information, and Communication.
9. The International Journal of Applied Aviation Studies.
10. The MATLAB© User Community Applications list.
11. The USAF Center for Systems Engineering.

Recommendation for Future Research

This research is the impetus for multiple continuation research efforts. Further research has already begun to investigate the ideal similarity weighting for use in Method 1. This involves reserving the mishap histories of specific aircraft as truth data and studying how well the method would have predicted the program's requirement priorities based on the remaining data. The DoD plans to replace DoD-HFACS with an official DoD Human Error Taxonomy in FY 2010. Work will be needed to update the RELAAy tool so that it remains relevant for future mishap data analysis. Research is ongoing to gather community of interest feedback of the proposed HFACS to HSI domain mapping.

Debates over the future of automation need a more quantitative approach. Applying Method 2 to those studies, whether regarding UAVs or industrial automation, would be a valuable contribution to the discussion. This applied research would quantify the performance tradeoffs of replacing humans in critical activities.

Research is needed into the expanded application of Method 3. The tool that implements Method 3 has been downloaded and used in a variety of application including portable thumb stick device design. What is needed is a through human subjects test to compare the performance of input devices designed using Method 3 and the performance of those using traditional control system design techniques. The application of this design method for haptic control and force feedback devices would also be of interest to the UI design community. In addition, Method 3 assumed that the user population could be represented by median human operator performance parameters. Further research is needed to explore the effectiveness of tailored parameters derived from sampling targeted user groups.

Further research is currently ongoing to study the applicability Method 4 for other display device scenarios; UAS portable display layouts in particular. This has led to some possible extensions of the method using cluster analysis of the graph to design even better layouts. Another research effort is studying how to expand the method by removing the constraint of fixed page cardinality. If this is possible, Method 4 would have even greater utility as a display layout design tool. An additional potential area of research is to evaluate the use of other human performance coefficients for other display hardware configurations such as mouse selections or haptic control devices. Finally, like the previous method, Method 4 assumed that the user population could be represented by median human operator performance parameters. Further research is needed to explore the effectiveness of tailored parameters derived from sampling targeted user groups.

Final Conclusions

The domains of human systems integration have only recently been formalized into a specific area of emphasis in systems engineering. Though its criticality has long been recognized, attempts to engineer HSI have been foiled by a lack of effective methods and tools. This dissertation research has laid forth a broadly applicable methodology applied through methods that better support the SE technical processes. It has demonstrated that: key insight can be gained early in the requirements development process if legacy system mishaps are properly studied; logical analysis is improved with quantitative function allocation; and, better user interfaces can be considered during the design solution process with methods and criteria that enable early objective analysis.

Appendix A: RELAAy Tool Outputs

This appendix contains the outputs of the Requirements Elicitation through Legacy Accident Analysis (RELAAy) Tool as applied to the MQ-X program.

Table A1. Consolidated HFACS Code Data

Data grouped by HFACS Code

of codes: 147
used: 120

HFACS Nanocode	Title	Occurences/ 5 yr	Severity rating	Risk Rating	Similarity Factor	Significance		Std Dev Severity	Std Dev Similarity
AE101	Inadvertent Operation	7	2.71	5.1	0.62	3.13		0.45	0.10
AE102	Checklist Error	8	2.75	5.3	0.65	3.44		0.43	0.07
AE103	Procedural Error	8	3.00	5.8	0.65	3.75		0.00	0.07
AE104	Overcontrol/Undercontrol	9	3.00	5.9	0.60	3.55		0.00	0.09
AE105	Breakdown in Visual Scan	4	3.00	4.9	0.57	2.79		0.00	0.11
AE106	Inadequate Anti-G Straining Maneuver	2	3.00	4.0	0.47	1.85		0.00	0.00
AE201	Risk Assessmnent – During Operation	16	2.88	6.4	0.66	4.25		0.33	0.05
AE202	Task Misprioritization	10	3.00	6.1	0.57	3.45		0.00	0.10
AE203	Necessary Action – Rushed	3	2.67	4.0	0.68	2.72		0.47	0.00
AE204	Necessary Action – Delayed	15	3.00	6.6	0.65	4.27		0.00	0.05
AE205	Caution/Warning – Ignored	2	2.50	3.3	0.68	2.25		0.50	0.00
AE206	Decision Making During Operation	7	2.86	5.3	0.62	3.30		0.35	0.10
AE301	Error due to Misperception	5	3.00	5.2	0.62	3.23		0.00	0.08
AV001	Violation - Based on Risk Assessment	0	--	0.0	--	0.00		0.00	0.00
AV002	Violation - Routine/Widespread	1	2.00	2.0	0.68	1.39		0.00	0.00
AV003	Violation - Lack of Discipline	8	2.88	5.5	0.66	3.68		0.33	0.01
PE101	Vision Restricted by Icing/Windows Fogged/Etc.	1	3.00	3.1	0.68	2.08		0.00	0.00
PE102	Vision restricted by Meteorological Conditions	10	3.00	6.1	0.66	4.02		0.00	0.01
PE103	Vibration	0	--	0.0	--	0.00		0.00	0.00
PE104	Vision Restricted in Workplace by Dust/Smoke/Etc.	1	3.00	3.1	0.65	2.00		0.00	0.00
PE105	Windblast	1	3.00	3.1	0.47	1.43		0.00	0.00
PE106	Thermal Stress – Cold	0	--	0.0	--	0.00		0.00	0.00
PE107	Thermal Stress – Heat	0	--	0.0	--	0.00		0.00	0.00
PE108	Manuevering Forces – In-Flight	10	2.10	4.2	0.66	2.79		0.30	0.06
PE109	Lightning of Other Aircraft/Vehicle	0	--	0.0	--	0.00		0.00	0.00
PE110	Noise Interference	1	3.00	3.1	0.68	2.08		0.00	0.00
PE111	Brownout/Whiteout	0	--	0.0	--	0.00		0.00	0.00
PE201	Seating and Restraints	5	2.40	4.1	0.63	2.61		0.49	0.08
PE202	Instrumentation and Sensory Feedback Systems	6	3.00	5.4	0.64	3.48		0.00	0.08
PE203	Visibility Restrictions	4	3.00	4.9	0.67	3.28		0.00	0.01
PE204	Controls and Switches	5	3.00	5.2	0.63	3.26		0.00	0.08
PE205	Automation	2	3.00	4.0	0.47	1.85		0.00	0.00
PE206	Workspace Incompatible with Human	0	--	0.0	--	0.00		0.00	0.00
PE207	Personal Equipment Interference	1	3.00	3.1	0.65	2.00		0.00	0.00
PE208	Communications – Equipment	2	3.00	4.0	0.68	2.70		0.00	0.00
PC101	Inattention	8	3.00	5.8	0.65	3.75		0.00	0.07
PC102	Channelized Attention	37	2.97	7.7	0.63	4.84		0.16	0.08
PC103	Cognitive Task Oversaturation	9	3.00	5.9	0.58	3.44		0.00	0.10
PC104	Confusion	8	3.00	5.8	0.65	3.75		0.00	0.07
PC105	Negative Transfer	6	2.83	5.1	0.68	3.47		0.37	0.00
PC106	Distraction	7	2.71	5.1	0.65	3.29		0.45	0.07
PC107	Geographic Misorientation (Lost)	0	--	0.0	--	0.00		0.00	0.00
PC108	Checklist Interference	0	--	0.0	--	0.00		0.00	0.00
PC201	Pre-Exisiting Personality Disorder	0	--	0.0	--	0.00		0.00	0.00
PC202	Pre-Existing Psychological Disorder	0	--	0.0	--	0.00		0.00	0.00
PC203	Pre-Existing Psychosocial Disorder	4	2.75	4.5	0.68	3.03		0.43	0.00
PC204	Emotional State	1	3.00	3.1	0.68	2.08		0.00	0.00
PC205	Personality Style	3	3.00	4.5	0.66	2.97		0.00	0.01
PC206	Overconfidence	13	3.00	6.4	0.64	4.07		0.00	0.07
PC207	Pressing	0	--	0.0	--	0.00		0.00	0.00
PC208	Complacency	19	2.68	6.2	0.67	4.15		0.46	0.01
PC209	Inadequate Motivation	2	2.00	2.6	0.68	1.80		0.00	0.00
PC210	Misplaced Motivation	0	--	0.0	--	0.00		0.00	0.00
PC211	Overaggressive	2	3.00	4.0	0.68	2.70		0.00	0.00
PC212	Excessive Motivation to Succeed	1	3.00	3.1	0.47	1.43		0.00	0.00
PC213	Get-Home-It is/Get-There-It is	3	3.00	4.5	0.68	3.06		0.00	0.00
PC214	Response Set	11	3.00	6.2	0.65	4.00		0.00	0.06
PC215	Motivational Exhaustion (Burnout)	1	3.00	3.1	0.68	2.08		0.00	0.00
PC301	Effects of G-forces (G-LOC, etc.)	4	3.00	4.9	0.52	2.53		0.00	0.09
PC302	Prescribed Drugs	0	--	0.0	--	0.00		0.00	0.00
PC303	Operational Injury/Illness	0	--	0.0	--	0.00		0.00	0.00
PC304	Sudden Incapacitation/Unconsciousness	0	--	0.0	--	0.00		0.00	0.00
PC305	Pre-Exisiting Physical Illness/Injury/Deficit	6	3.00	5.4	0.67	3.64		0.00	0.01
PC306	Physical Fatigue (Overexertion)	3	3.00	4.5	0.54	2.41		0.00	0.10
PC307	Fatigue – Physiological/Mental	14	3.00	6.5	0.65	4.21		0.00	0.07
PC308	Circadian Rhthym Desynchrony	7	3.00	5.6	0.68	3.81		0.00	0.00
PC309	Motion Sickness	1	3.00	3.1	0.68	2.08		0.00	0.00
PC310	Trapped Gas Disorder	0	--	0.0	--	0.00		0.00	0.00
PC311	Evolved Gas Disorder	0	--	0.0	--	0.00		0.00	0.00
PC312	Hypoxia	1	3.00	3.1	0.68	2.08		0.00	0.00

HFACS Nanocode	Title	Occurences/ 5 yr	Severity rating	Risk Rating	Similarity Factor	Significance		Std Dev Severity	Std Dev Similarity
PC316	Physical Task Oversaturation	0	--	0.0	--	0.00		0.00	0.00
PC401	Learning/Ability/Rate	0	--	0.0	--	0.00		0.00	0.00
PC402	Memory Ability/Lapses	0	--	0.0	--	0.00		0.00	0.00
PC403	Anthropometric/Biomechanical Limitations	1	2.00	2.0	0.68	1.39		0.00	0.00
PC404	Motor Skills/Coordination or Timing Deficiency	1	3.00	3.1	0.68	2.08		0.00	0.00
PC405	Technical/Procedural Knowledge	2	3.00	4.0	0.68	2.70		0.00	0.00
PC501	Illusion – Kinesthetic	1	3.00	3.1	0.68	2.08		0.00	0.00
PC502	Illusion – Vestibular	3	3.00	4.5	0.68	3.06		0.00	0.00
PC503	Illusion – Visual	1	3.00	3.1	0.68	2.08		0.00	0.00
PC504	Misperception of Operational Conditions	20	3.00	7.0	0.60	4.18		0.00	0.09
PC505	Misinterpreted/Misread Instrument	2	3.00	4.0	0.68	2.70		0.00	0.00
PC506	Expectancy	11	3.00	6.2	0.63	3.88		0.00	0.08
PC507	Auditory Cues	2	2.50	3.3	0.57	1.89		0.50	0.11
PC508	Spatial Disorientation 1 Unrecognized	15	3.00	6.6	0.65	4.31		0.00	0.05
PC509	Spatial Disorientation 2 Recognized	3	3.00	4.5	0.68	3.06		0.00	0.00
PC510	Spatial Disorientation 3 Incapacitating	2	3.00	4.0	0.68	2.70		0.00	0.00
PC511	Temporal Distortion	11	3.00	6.2	0.67	4.12		0.00	0.01
PP101	Crew/Team Leadership	0	--	0.0	--	0.00		0.00	0.00
PP102	Cross-Monitoring Performance	2	3.00	4.0	0.68	2.70		0.00	0.00
PP103	Task Delegation	3	3.00	4.5	0.67	3.01		0.00	0.01
PP104	Rank/Position Authority Gradient	1	3.00	3.1	0.68	2.08		0.00	0.00
PP105	Assertiveness	0	--	0.0	--	0.00		0.00	0.00
PP106	Communicating Critical Information	1	3.00	3.1	0.68	2.08		0.00	0.00
PP107	Standard/Proper Terminology	3	3.00	4.5	0.61	2.73		0.00	0.10
PP108	Challenge and Reply	0	--	0.0	--	0.00		0.00	0.00
PP109	Mission Planning	5	2.80	4.8	0.66	3.19		0.40	0.01
PP110	Mission Briefing	2	3.00	4.0	0.57	2.27		0.00	0.11
PP111	Task/Mission-In-Progress Re-Planning	1	3.00	3.1	0.65	2.00		0.00	0.00
PP112	Miscommunication	3	2.67	4.0	0.68	2.72		0.47	0.00
PP201	Physical Fitness	0	--	0.0	--	0.00		0.00	0.00
PP202	Alcohol	2	3.00	4.0	0.68	2.70		0.00	0.00
PP203	Drugs/Supplements/Self Medication	0	--	0.0	--	0.00		0.00	0.00
PP204	Nutrition	1	3.00	3.1	0.47	1.43		0.00	0.00
PP205	Inadequate Rest	2	3.00	4.0	0.57	2.27		0.00	0.11
PP206	Unreported Disqualifying Medical Condition	0	--	0.0	--	0.00		0.00	0.00
SI001	Leadership/Supervision/Oversight Inadequate	30	2.40	6.0	0.69	4.11		0.49	0.10
SI002	Supervision – Modeling	2	3.00	4.0	0.84	3.32		0.00	0.16
SI003	Local Training Issues/Programs	44	2.55	6.8	0.73	4.99		0.50	0.12
SI004	Supervision – Policy	9	2.33	4.6	0.78	3.61		0.47	0.15
SI005	Supervision – Personality Conflict	1	2.00	2.0	0.68	1.39		0.00	0.00
SI006	Supervision – Lack of Feedback	1	3.00	3.1	0.99	3.04		0.00	0.00
SP001	Ordered/Led on Mission Beyond Capability	2	2.50	3.3	0.82	2.72		0.50	0.17
SP002	Crew/Team/Flight Makeup/Composition	9	2.44	4.8	0.82	3.95		0.50	0.16
SP003	Limited Recent Experience	10	2.50	5.1	0.71	3.58		0.50	0.10
SP004	Limited Total Experience	23	2.52	6.0	0.77	4.60		0.50	0.16
SP005	Proficiency	24	2.29	5.5	0.69	3.79		0.45	0.13
SP006	Risk Assessment – Formal	3	2.67	4.0	0.67	2.68		0.47	0.01
SP007	Authorized Unnecessary Hazard	1	3.00	3.1	0.65	2.00		0.00	0.00
SF001	Personnel Management	1	3.00	3.1	0.68	2.08		0.00	0.00
SF002	Operations Management	10	2.80	5.7	0.74	4.20		0.40	0.13
SV001	Supervision – Discipline Enforced	12	2.67	5.6	0.76	4.24		0.47	0.14
SV002	Supervision – De facto Policy	8	2.50	4.8	0.65	3.11		0.50	0.07
SV003	Directed Violation	1	2.00	2.0	0.99	2.03		0.00	0.00
SV004	Currency	2	2.00	2.6	0.68	1.80		0.00	0.00
OR001	Air Traffic Control Resources	2	3.00	4.0	0.68	2.70		0.00	0.00
OR002	Airfield Resources	15	2.53	5.6	0.70	3.90		0.50	0.08
OR003	Operator Support	1	2.00	2.0	0.68	1.39		0.00	0.00
OR004	Acquisition Polies/Design Processes	35	2.57	6.6	0.73	4.85		0.49	0.13
OR005	Attrition Policies	2	2.50	3.3	0.84	2.76		0.50	0.16
OR006	Accession/Selection Policies	1	3.00	3.1	0.68	2.08		0.00	0.00
OR007	Personnel Resources	10	2.30	4.7	0.80	3.74		0.46	0.15
OR008	Informational Resources/Support	3	2.67	4.0	0.89	3.55		0.47	0.15
OR009	Financial Resources/Support	1	2.00	2.0	0.68	1.39		0.00	0.00
OC001	Unit/Organizational Values/Culture	19	2.58	5.9	0.75	4.44		0.49	0.15
OC002	Evaluation Promotion/Upgrade	3	2.33	3.5	0.71	2.49		0.47	0.22
OC003	Perceptions of Equipment	12	2.50	5.3	0.73	3.83		0.50	0.12
OC004	Unit Msn/AC/ Vehicle/Equip Change or Unit Deactivatio	3	2.00	3.0	0.68	2.04		0.00	0.00
OC005	Organizational Structure	1	3.00	3.1	0.99	3.04		0.00	0.00
OP001	Ops Tempo/Workload	22	2.45	5.8	0.79	4.60		0.50	0.15
OP002	Program and Policy Risk Assessment	14	2.79	6.0	0.78	4.72		0.41	0.20
OP003	Procedural Guidance/Publications	100	2.58	7.8	0.72	5.63		0.49	0.13
OP004	Organizational Training Issues/Programs	23	2.57	6.1	0.73	4.46		0.50	0.15
OP005	Doctrine	4	2.50	4.1	0.76	3.08		0.50	0.14
OP006	Program Oversight/Program Management	8	2.50	4.8	0.72	3.46		0.50	0.10
		Averages:	2.53	3.6	0.75	2.45			

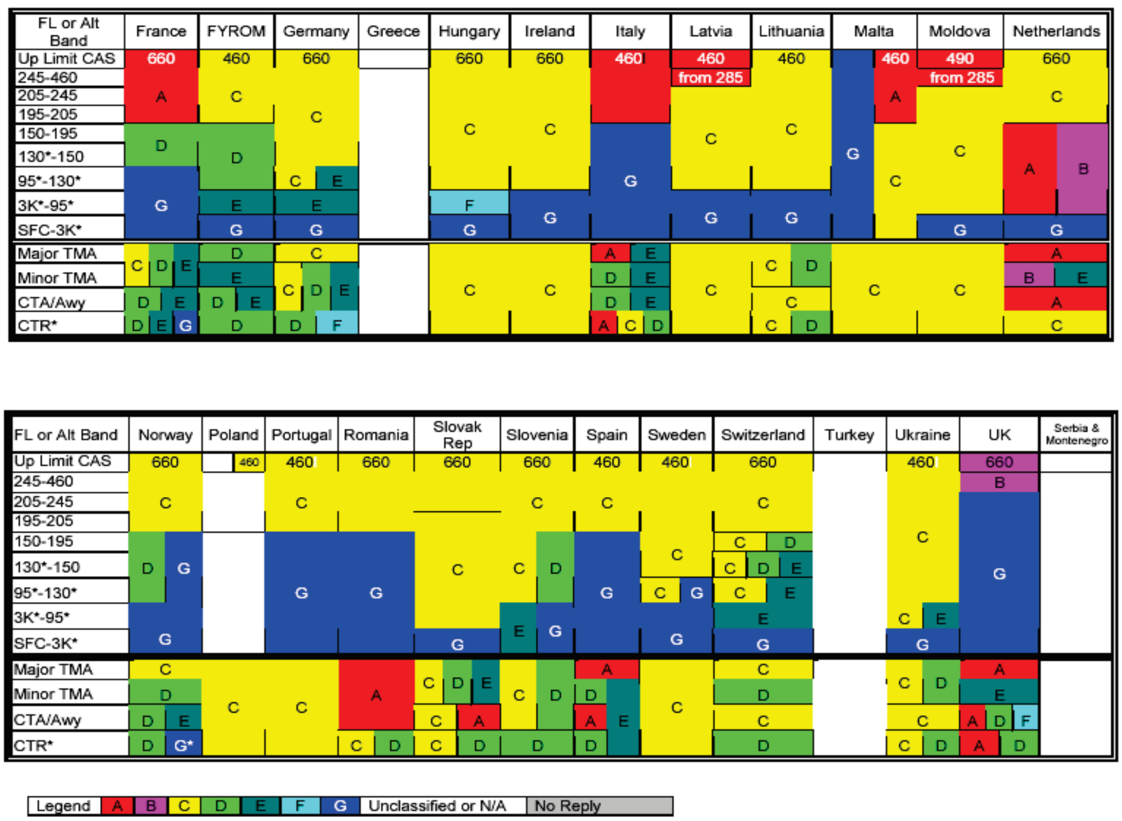
Table A2: MQ-X Top Twenty List

Program Overview of all Domains

HFACS Nanocode	Title	Significance for MQ-X	Manpower	Personnel	Training	Human Factors	Safety	Health	Habitability	Occurences/ 5 yr	Severity rating	Std Dev Severity	Similarity Factor	Std Dev Similarity
OP003	Procedural Guidance/Publications	5.63			T					100	2.58	0.49	0.72	0.13
SI003	Local Training Issues/Programs	4.99			T					44	2.55	0.50	0.73	0.12
OR004	Acquisition Polies/Design Processes	4.85					S			35	2.57	0.49	0.73	0.13
PC102	Channelized Attention	4.84			T	HF				37	2.97	0.16	0.63	0.08
OP002	Program and Policy Risk Assessment	4.72					S			14	2.79	0.41	0.78	0.20
OP001	Ops Tempo/Workload	4.60	M	P		HF				22	2.45	0.50	0.79	0.15
SP004	Limited Total Experience	4.60		P	T					23	2.52	0.50	0.77	0.16
OP004	Organizational Training Issues/Programs	4.46			T					23	2.57	0.50	0.73	0.15
OC001	Unit/Organizational Values/Culture	4.44		P	T					19	2.58	0.49	0.75	0.15
PC508	Spatial Disorientation 1 Unrecognized	4.31				HF				15	3.00	0.00	0.65	0.05
AE204	Necessary Action – Delayed	4.27		P	T	HF				15	3.00	0.00	0.65	0.05
AE201	Risk Assessmnent – During Operation	4.25		P	T					16	2.88	0.33	0.66	0.05
SV001	Supervision – Discipline Enforced	4.24					S			12	2.67	0.47	0.76	0.14
PC307	Fatigue – Physiological/Mental	4.21	M	P						14	3.00	0.00	0.65	0.07
SF002	Operations Management	4.20					S			10	2.80	0.40	0.74	0.13
PC504	Misperception of Operational Conditions	4.18				HF				20	3.00	0.00	0.60	0.09
PC208	Complacency	4.15		P						19	2.68	0.46	0.67	0.01
PC511	Temporal Distortion	4.12				HF				11	3.00	0.00	0.67	0.01
SI001	Leadership/Supervision/Oversight Inadequate	4.11	M	P	T					30	2.40	0.49	0.69	0.10
PC206	Overconfidence	4.07		P	T					13	3.00	0.00	0.64	0.07

Appendix B: Function Allocation Performance Calculations

This appendix contains the supporting table and figure for Method 2 discussed in Chapter 5.



3K* = FL55/ Alt 1,000 /1,500 /2,000 /2,500 /3,000 /3,500 /5,000 (ft AGL/AMSL)
95* = FLs 75/95/100/ Alt 7,500
130* = FLs 115/125/130/135
CTR* = CTR/ Aerodrome Zone
G* = G or G with special conditions

Figure B1. European Airspace Designation by Country
(Eurocontrol Website, 2007)

Table B1: Proposed UAV Classification

Class	**Description**	**Example**	**Operating Altitude [ft]**	**Range Restriction**	**Cruise Speed [kts]**	**Max Weight [lb] or [kg]**	
3	High Altitude, Long Endur. (HALE); Stratospheric	Global Hawk, Aerosonde	>35,000	Beyond LOS	>250	>4500	>2000
2	Medium Altitude, Long Endur. (MALE);	Eagle, Predator	35,000	Beyond LOS	< 250	4500	2000
1	Short range	SIVA, ALO, Pioneer	18,000	LOS only	< 150	1100	500
0	Mini, and micro	R-Max, MicroStar	<1,000	LOS only	< 100	45	25

Appendix C: Input Device Optimization Model

This appendix contains the block diagrams of the Simulink® model used in the algorithm of Method 3 as described in Chapter 6. This tool is available for download at The Mathworks® User Community website: www.mathworkscentral.com.

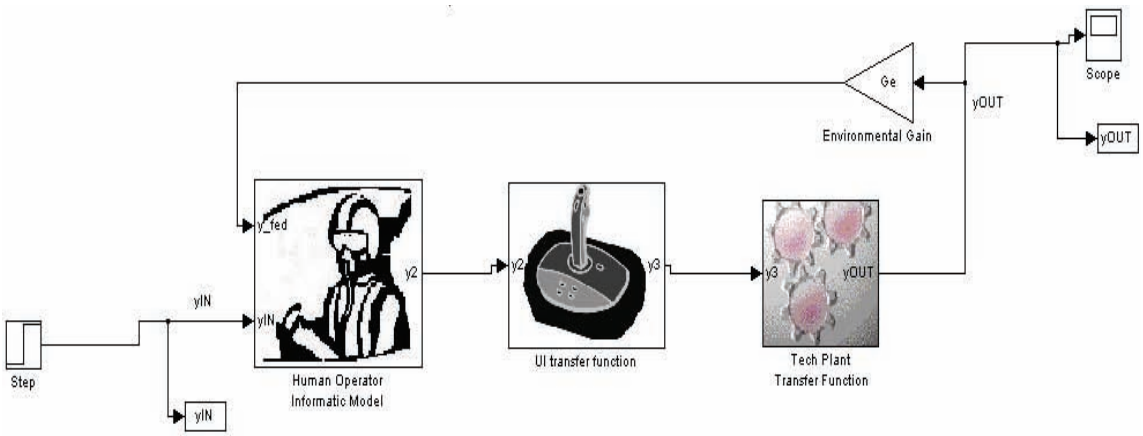


Figure C1. System Model in Simulink®

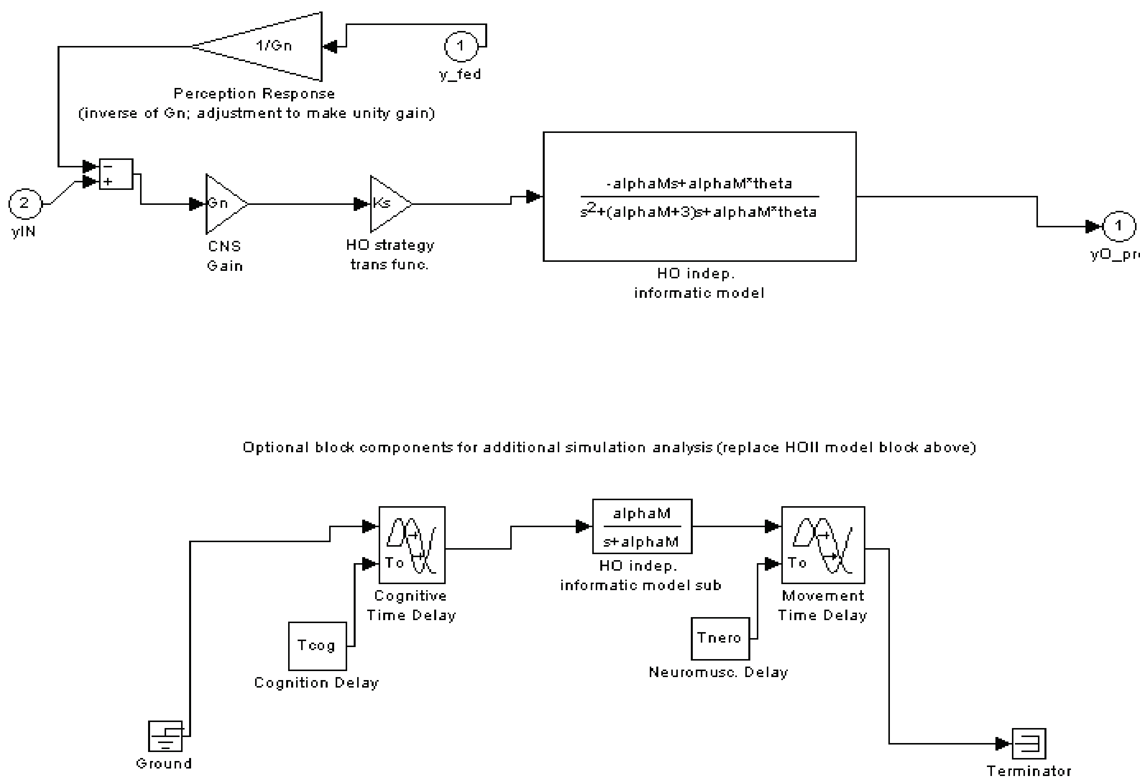


Figure C2. Human Operator Informatic Model Block in Simulink®

Appendix D: Display Layout Design Method Supporting Information

This appendix contains foundational information for Method 4, Display Layout Design, as described in Chapter 7. The information contained in this appendix was published in two IEEE conference papers:

D1: IEEE DHMS International Conference, 2008

Co-authored with: J. Colombi, D. Jacques, and R. Hill

D2: IEEE SMC International Conference, 2009

Co-authored with: J. Colombi, D. Jacques, R. Hill, and J. Miller

Appendix D1: Improved UI Design through Graph-Theoretic Modeling

Introduction

Since the inception of modern computing, researchers have studied how humans and computers interact. As computers became more ubiquitous, programmers recognized the need to identify best practices for the user interface. This body of knowledge has matured and expanded in perspective and is now referred to as the study of human-computer interaction (HCI) [1]. In aviation and military environments the demand for effective HCI is paramount. This is motivating engineers to develop better ways to design user interfaces (also called PVI--pilot-vehicle interfaces, or more generally, HMI--human-machine interfaces) [2]. This paper shows how we can draw from the established mathematics of graph theory to create metrics for the tradeoff analysis of interface design.

Background

Formerly, cockpit displays were grouped into prioritized subdivisions of the user's field of view. For instance, the classic T-configuration (arrangement of the airspeed indicator, attitude indicator, altimeter, and directional gyro) became a de facto standard for the primary visual area. Information deemed less critical was relegated more to the periphery. Controls were categorized similarly, and the final design was an allocation of the limited visual "real estate" between critical displays and high priority controls [3].

Avionics designers are now sufficiently comfortable with the reliability, functionality and affordability of flatscreen displays to incorporate "glass cockpits" in current aircraft. That is, the number of mechanical switches and dials (called "steam gauges" by pilots) has been greatly reduced and more information is consolidated on central multi-function displays (MFD). Now information can be combined and superimposed in novel ways

FIGURE 1. The F-35 cockpit (prototype) shows most dedicated gauges and switches replaced by two large touchscreen multi-function displays [4].

that reduce pilot workload. For example, the situational information from all sensors and data links (i.e. weather radar, traffic avoidance radar, transponder, etc.) can be superimposed onto navigation information, such as a moving map display, to form a fully integrated horizontal situation display (HSD).

Current production aircraft, such as the Boeing 777 and the F-22, use flatscreen MFDs with bezel buttons around the edges. These are called "soft buttons" as their functions can be tailored to the particular screen being displayed. Fig. 1 shows a next generation cockpit, which is already in development for the F-35 Joint Strike Fighter. Designers have chosen to make the MFD even larger and with touchscreen capability. Touchscreen MFDs are being proposed for future unmanned aerial vehicle (UAV) control as well. This design eliminates the need for peripheral buttons as the user can manipulate options directly on the screen [4]. Interface designers call this capability *direct manipulation*. It yields a more intuitive interface because it emulates real life actions [1]. In addition to the touchscreen, the F-35 PVI includes:

- a helmet-mounted display,
- backup gauges,
- 3D sound annunciators,
- hands on throttle and stick (HOTAS) buttons, and
- programmable voice command recognition.

Cockpit and UAV ground control station (GCS) design traditionally focused on the design of controls and displays as independent problems; designers made switches/buttons for the controls and gauges for the displays. Technological advances, such as those listed above, have changed the modern HCI design problem by facilitating the fusion of controls and displays; however, for functional analysis purposes, it is still convenient to maintain the two groups. In viewing the interface from the operator's perspective, the interface input components are collectively called the "controls". This includes switches, buttons, soft buttons, selectable icons, and even voice recognition apparatus. The interface output components are collectively called the "displays". This not only includes panels, gauges, lights, and labels, but audible alarms tactile sensors and voice messages as well.

Cockpit design begins with specifications that list all necessary information and configuration requirements [3]. The task of the designer is to find the best layout that satisfies these specifications. HCI design principles recommend a *context-sensitive* (or *context-aware*) interface design paradigm where the informatic relationships are based on what is needed for a given operational activity thread [1]. For example, in the landing phase of flight, operators must draw from multiple otherwise-unrelated sources of information. They must correctly update their mental model by continuously scanning critical information and selectively skipping what is not important for the immediate task. This learned skill is called a proper "cross check", and it is an accommodation of humans to machine limitations. A context-aware configuration would sense the present activity

and tailor the availability of information and selection options.

Technology has freed designers from dedicated displays; designers now need to free themselves from a dedicated display mindset. Experimental psychologists have correlated necessary information by phase of flight [5]. Designers can now use this foundational data, combined with graph-theoretic modeling, in cockpit design. When converted to a graph-theoretic problem, finding the best layout, given the specifications, becomes an intuitive activity.

Design Challenge

The glass cockpit gives developmental engineers new design freedom never before experienced; however, like any frontier, the lack of constraints increases the decision complexity and highlights the need for sound principles. Current best practices have made HCI explicit in system design by maturing the requirements engineering and task analysis processes to include the study of interaction patterns [6]. More improvement is necessary, however.

One area for improvement is how to find the delicate balance between conflicting design criteria such as: (1) present an intuitive structure and (2) streamline the layout for rapid task execution. In addition, some information may be of very low priority most of the time, but very critical in a rare but dire emergency. Prioritization schemes that do not accommodate all requirements will lead to systems that fail to achieve the desired level of usability and safety. As we shall demonstrate, tradeoff analyses transform readily into the graph theory domain.

Improvement is also needed in performing objective design comparison. This is currently done heuristically or through costly user testing. Analytic metrics are very desirable because they simplify the analysis and require much less effort. When graph-theoretic models are used, the layouts can be analyzed using the mature theorems of graph theory. The following sections will explain how a graph is derived from a design specification and how to analyze such a model. The example in Section VI demonstrates the process. The analysis was performed using the Matlab® graph theory toolbox [7] and the graphics were generated in Graphviz Neato® [8].

Model Construction

The first step of graph-theoretic analysis is to correctly construct a graph that models the proposed design layout. A graph, in the strict mathematical sense, consists of a set of *vertices* (or nodes) and a set of *edges*, with each edge being the connection between a specific vertex pair. Visually, a graph is represented by a set of points with connective lines between them. Mathematically, a graph can be expressed in several matrix forms. The most intuitive is an *adjacency matrix* of ones and zeros. For each i,j element equal to one, there exists an edge between vertex i and vertex j. For a graph to have real world meaning, we must precisely define the vertices and edges in the model.

Displays Categorization

Criticality Level	Constraints	Examples
Continuous	Must be always present	Flight data: airspeed, altitude, attitude
1	Must be accessible by no more than 1 selections	Critical systems status: hydraulics, electrical, engine temperature
2	Must be accessible by no more than 2 selections	Secondary systems: navigation or communications equipment
Normal	Prioritized by frequency of use	All other info/data

Vertices

Let each vertex in a graph model be defined as one interface output configuration. The information presented in each vertex is a unique subset of all information available. The term "information" is used here specifically for data that has been processed into a useful form. An interface output is a specific set of information items that is independent of the many ways that information can be represented. For example, altitude is an information item that will be included in multiple vertices of a cockpit design: on a setup screen, the altitude may be a text box; on an instrument approach screen the altitude may be a scrolling tape display; and in terrain following mode, altitude may be relayed by voice annunciators ("Altitude! Pull up.").

Table 1 shows an example of how information might be grouped into displays and categorized using traditional design methods. Displays with a high criticality level are constraints on the solution. Displays in the normal category would be organized by prioritization schemes. This paper will use the term "screen" when specifically referring to user interface vertices as it makes the article more readable, but the vertex set includes the total presentation of information including audible, haptic, and tactile. A similar mapping of displayed information is used to model webpage interconnections so that

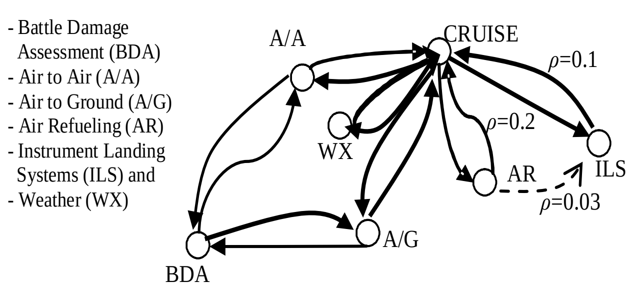


FIGURE 2. A sample graph-theoretic model of a user interface. The nodes reflect screens for mission tasks. Sample values for ρ are for use in Sec. V.

mathematicians can analyze the “diameter” of the World Wide Web [9].

Edges

Let the edges of a graph model be specific interface inputs from the human that initiate a change of display state. Interface inputs are distinct from task-related inputs, such as data entry or weapon guidance inputs, but in this paper, we will simply use “input” when the context is clear. Table 2 shows some example categorizations by input type and the corresponding constraints on the design solution.

Graph Description

The edges, as defined, are unidirectional transitions. Therefore, in mathematical terms, the model is a directed graph or *digraph*; however, most real world designs allow for reversibility of action. In fact, this is one of the golden rules of interface design identified in [1]. Though the following tools are valid regardless, a proper design will, in general, only consist of complimentary edge pairs. This is modeled as a bidirectional graph.

A complete model may contain “multiple edges” and/or “loops”. Multiple edges arise in a case where more than one input causes the same screen transition (e.g. a combination of a voice command, shortcut key, or a menu button), and loops represent an input that does not change the display state (e.g. zoom or declutter). For layout analysis, multiple edges and loops are not included and the underlying simple graph is used. This results in a symmetric adjacency matrix with zeros on the diagonal.

This paper will use the term “HCI-defined” to refer to a graph *G* that has been built in accordance with the definitions of this paper and is therefore a representative model of a valid interface layout. A simple valid graph is shown in Fig. 2.

Graph-Theoretic Analysis

Once the proposed layout has been graphically defined, we can analyze the resulting digraph. The following italicized terminology is a list of novel terms specifically applied to interface design, but they have been defined in a way consistent with current mathematical literature [10]. These graph parameters form a collection of tools for

Controls Categorization

Type	Description	Examples
Dedicated	Input dedicated for one action (e.g. specialized buttons, voice)	Radio transmit, chaff, flare, arm/unarm
Virtual	Input that is context-sensitive (soft buttons, selectable items on touchscreen)	Selection opens a new window
Screen Modify	Input that does not cause a screen transition	Zoom, declutter

requirements verification and comparative design analysis.

Degree

For any screen represented by vertex v in an HCI-defined graph G, the following definitions are used:

- The *indegree* $d^{-}(v)$ is the number of edges which have v as the head.
- The *outdegree* $d^{+}(v)$ is the number of edges which have v as the tail.
- The minimum and maximum indegree of a graph G are $\delta^{-}(G)$ and $\Delta^{-}(G)$, respectively; for outdegree we use $\delta^{+}(G)$ and $\Delta^{+}(G)$, respectively.

Using these definitions, the following useful bounds on the model can be established.

Note that the vertex outdegree $d^{+}(v)$ defines the number of interface inputs (controls) that must be programmed for a specific screen. Using the graph theory degree sum formula [10]:

$$\sum_{v \in V(G)} d^{+}(v_i) = \sum_{v \in V(G)} d^{-}(v_i) = |E(G)|$$

where:

$|E(G)|$ is the total number in the edge set of G.

This provides a metric on programming effort for proposed layouts.

There is no bound on the indegree of any specific vertex since a screen is unaffected by the number of ways users are able to transition to that screen; however, in general, the maximum outdegree, $\Delta^{+}(G)$, of any specific vertex will be limited by physical constraints. Since $\Delta^{+}(G)$ represents the maximum number of interface inputs programmed for a specific screen in the design, it is constrained by the maximum number of on-screen selectable objects allowed.

A valid layout cannot have isolated points, so the lower bounds on minimum degrees are $\delta^{-}(G) \geq 1$, $\delta^{+}(G) \geq 1$. Also, a valid layout cannot have dead ends, so the lower bound on the total program effort is:

$$\sum d^{+}(v_i) \geq n$$

where:

n is the cardinality (size) of the vertex set.

The above bounds affect interface tradeoff analysis such as the size of a voice command register or the mix between dedicated switches and soft buttons.

Distance

In graph theory, path length is defined as the number of edges on a path between two

vertices, and distance is defined as the length of the shortest path. For an interface layout this corresponds to the number of inputs required to move from one screen to another.

A more interesting measure of distance is achieved if each edge is weighted by the expected time delays associated with each screen change. Time delays are the result of human decision making, activation (movement or articulation), transmission, and processing. The processing delay is obtained from network analysis, but is usually only significant in teleoperation scenarios (e.g. geographically-separated control of unmanned vehicles). Estimates for human decision making delays are obtained from the literature on information theory and numerous experiments with human subjects. According to the Hick-Hyman Law, assuming a trained but still novice user, the time delay for decision making is logarithmically proportional to the number of elements in the decision list [11]. This permits the following definition:

The *weighted distance* from v_0 to v_k on an HCI-defined G, is written as $d_w(v_0,v_k)$, and defined as:

$$d_w(v_0,v_k)=\sum_{i=1}^{k}\left(t_i+\left(0.2+(0.15)\log_2\left(d^+(v_{i-1})\right)\right)\right) \tag{1}$$

where:

v_0 and v_k are any two arbitrary vertices,

t_i is a time delay constant associated with the i^{th} edge on the (v_0, v_k) minimum path, and

$d^+(v_{i-1})$ is the outdegree of the tail vertex of the i^{th} edge on the (v_0,v_k) minimum path

This yields a "distance" that is the expected time to traverse the given path based on the processing speed of the interface and the decision making speed of the operator. This metric is written as $d_{\text{w\_G}}(v_0,v_k)$ when the associated graph is not clear from the context.

The expected execution time of specific tasks using a proposed interface layout can be quantitatively analyzed using (1); however, the analysis of the overall design requires the following additional tools.

Eccentricity

Using the definition of weighted distance, we can now discuss the concept of eccentricity and use it to define specifications for proposed layouts:

The *eccentricity* of a vertex v_a in a graph is the maximum of the distances between v_a and all other vertices of the graph. Using (1), the *weighted in-eccentricity* is the maximum $d_{\text{w}}(v_b,v_a)$ for any v_b of G:

$$ecc_w^-(v_a)=\max_{v_b\in V(G)}\{d_w(v_b,v_a)\} \tag{2}$$

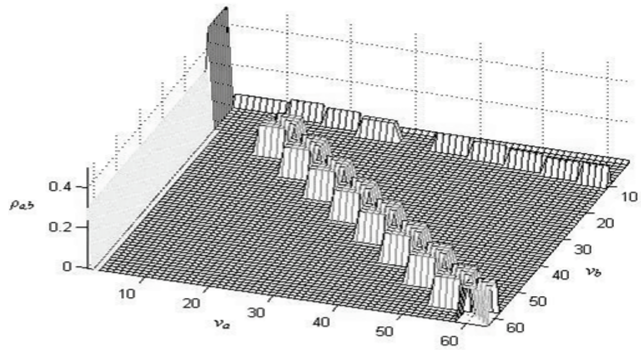


FIGURE 3: Graphical portrayal of affinity matrix used in the example.

Simply stated, this is a measure of the greatest separation, in transition time, that exists between a particular screen and any part of the rest of the graph. Since this is what developers truly want to control for critical information, $\mathrm{ecc}_w^-(v_a)$ is a better way to identify those constraints than traditional methods.

Diameter

Building on these concepts, we can discuss the weighted diameter for our interface design model. The diameter of a graph *dia*(*G*) is the maximum of all eccentricities. The *weighted in-diameter* of an HCI-defined *G*, written as $dia^-_w(G)$, is the maximum $d_w(v_0,v_k)$ of *G* which is the maximum of all weighted in-eccentricities:

$$dia_w^-(G) = \max_{v \in V(G)}(ecc_w^-(v)) \tag{3}$$

The size of the weighted diameter is a measure of the "long pole" in a proposed layout's task execution time. As such, it provides insight for design improvement.

It is beyond the scope of this paper, but numerous other graph parameters (such as radius, center, and periphery) can be used to gain insight into proposed interface designs.

Optimality Metrics

The most valuable contribution of graph-theoretic analysis is the ability to quantitatively compare proposed layouts. A simple sum of the distances between all vertices in *G* is called the Wiener Index *W(G)* [10], and provides a basic scoring of a model. A novel approach to user interface evaluation is to use a weighted distance summation:

$$D_w(G) = \frac{1}{n^2} \sum_{v_a, v_b \in V(G)} d_w(v_a, v_b) \tag{4a}$$

All indices used in this paper are normalized by the square of the number of vertices n^2. This is logical choice for normalizing the index because of the double summation over all vertices that occurs in the calculation. By normalizing, these indices can be used to make comparisons between graphs that do not have identical vertex sets. The index in (4a)

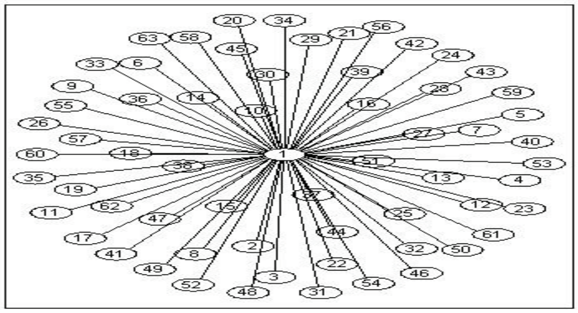


FIGURE 4. Layout 1: The "grand central" layout

yields a metric for evaluating the entire layout with respect to task execution time. This is useful for interface design, but minimizing (4a) does not guarantee that a layout is optimized for its mission, or even any specific task. For that, the metric must include input from its expected operational context. Designers can provide this in the form of an *affinity matrix* that captures the association of vertices with all others. To form the affinity matrix, first decompose the system's intended mission(s) into operational activities. Then, the likelihood of specific screen changes, such as accessing v_a from v_b, in the execution of all identified activities is a statistically-determined value $\rho_{a,b}$ (range: [0,1]). The affinity matrix is an n x n matrix of $\rho_{a,b}$ values. For example, in Fig. 2, $\rho_{\mathrm{AR,\ ILS}}=0.03 < \rho_{\mathrm{CRUISE,\ ILS}}=0.20$ which quantifies the fact that it is operationally less likely to transition directly to an ILS approach after air refueling than it is from cruise flight.

The total of any row (i.e.: all $\rho_{a,b}$ for a given v_b) must sum to 1.00. As shown in Fig. 3, in general, the affinity matrix will not be symmetric. A nonzero value on the diagonal represents the probability of repeating a completed task or performing a subsequent task that requires the same screen.

The affinity matrix used with (4a) creates the index:

$$D_{w.\rho}(G) = \frac{1}{n^2} \sum_{v_a, v_b \in V(G)} d_w(v_a, v_b) \cdot \rho_{a,b} \tag{4b}$$

Of all proposed layouts, the one with the smallest value of (4b) is guaranteed to have the lowest average task execution time. To see this, let G and H be HCI-defined for the same information output set. That is, $V(G)=V(H)$. Also, let $D_{w.\rho}(G) < D_{w.\rho}(H)$. Then, sampling the weighted distance between vertices v_a and v_b with a weighting of $\rho_{a,b}$ yields the expected values $E(d_{w\_G}(v_a,v_b))<E(d_{w\_H}(v_a,v_b))$. Thus, the expected task performance time is shorter in layout G than layout H. Therefore, (4b) gives an interface layout suitability metric; one that takes into account human decision making, the user interface, and the operational context. We call (4a) the HCI Index and (4b) the HCI Context-aware Index.

Example Application

The graph-theoretic analytical tools and indices can be applied to a wide variety of multi-modal operator station designs. The authors have found vignettes that show the

value of this approach in a wide array of user interfaces. For demonstration purposes, assume the following problem: the specifications for a new multi-role fighter require a user interface of n=63 screens with all reversible transitions and a time delay constant of t_i=1.0 ms for all interface inputs. Assume also that use cases and task analysis efforts have been completed and have supplied a complete list of $\rho_{a,b}$ estimates. Using the resultant affinity matrix, the vertices are sorted and numbered by common "cliques" as graphed in Fig. 3.

Three hypothetical configurations are proposed. In the first option, shown in Fig. 4, every possible display is a single input away from a central screen (such as the cruise screen). Some web-based applications use this actual layout. Intuitively, one can see that this design requires a very low number of inputs regardless of the task. This is confirmed analytically by observing the graph's small diameter, $dia(G)$=2. The obvious disadvantage is that, since all transitions are by way of a "grand central" screen the selection time will be relatively long on every transition.

In the second option, shown in Fig. 5, the designers have proposed a layout for minimum decision complexity. The screens are organized in a hierarchical structure according to the type of information provided. The resultant graph model has a "binary split" form. One can see that these first two designs represent opposing philosophies with respect to the number of interface inputs. The second option will require much more menu paging. This is confirmed by observing that the diameter, $dia(G)$=10, is much greater than the first option.

Lastly, the designers proposed a layout that was derived from a study of the task analysis information. This is shown in Fig. 6. With regard to graph diameter vs. decision complexity, this design is in the middle ground between the first two. As it can be seen, the data is organized into nine activity-based groups.

In order to evaluate the proposals, all three are first modeled in accordance with the process described earlier in this paper. Then, the tools of this paper are used to verify that all proposals are valid solutions (i.e.: all eccentricity limits and any other given constraints are met). The three proposals are then analyzed using the indices presented in

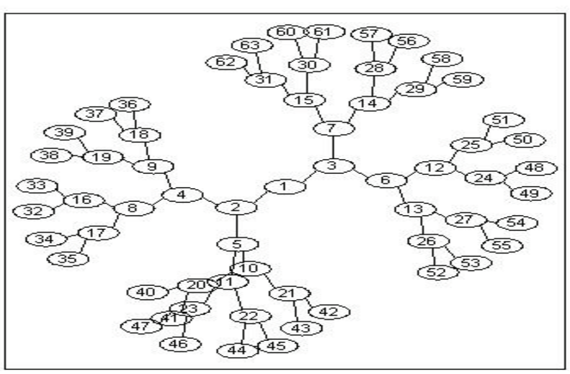


FIGURE 5. Layout 2: The "binary split" layout

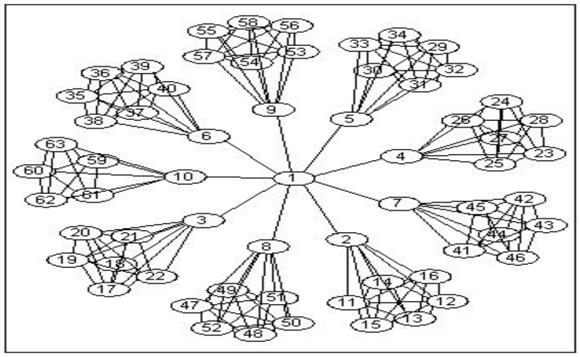


FIGURE 6. Layout 3: The "operational grouping" layout

this paper. The results are shown in Table 3. As it can be seen, the first layout possesses the lowest normalized Wiener Index and HCI Index. The second design, though logically organized and offering simple decision steps, ranks last in all metrics. Most significantly, however, the HCI Context-aware Index quantitatively identifies the third option as the best choice.

While the process shown in the previous example does not guarantee an optimal solution, the metrics could be used in constrained optimization problems. Expressed that way, the previous example would read:

Given:

- a specification-derived vertex set V(G),
- an operationally-relevant affinity matrix,
- a vector of constraints for $m \leq n$ critical vertices, and
- a solution space of l proposed interface designs, G_k.

$$\min\_ J = D_{w.\rho}(G_k) \quad \forall\, G_k \quad \{k : 1\; to\; l\}$$

$$\text{subject to: } ecc_w^-((v_{crit})_i) \leq b_i \quad i : 1\; to\; m$$

where:

v_{crit} are critical vertices, and

b_i are the respective constraints on the critical vertices.

TABLE 3
analysis Results for the example design problem

Layout Options	$W(G)/n^2$ [inputs]	$D_w(G)$ [s]	$D_{w.\rho}(G)$ [ks]
1. Grand Central	**1.93**	**1.25**	14.9
2. Binary split	6.48	2.68	33.1
3. Operational Grouping	3.34	2.09	**12.1**

Note: HCI Context-aware Index values multiplied by 1000.

Conclusion

As it has been shown, the use of graph theory improves the analysis of user interface designs. A transformation into a graph-theoretic model, in the manor described, will not only shed more light on design suitability, it can greatly reduce the effort required for quantitative evaluation. This facilitates the development of effective layouts for glass cockpits such as those discussed in Section II.

Future research should include user testing to validate that the given metrics generalize across the spectrum of user interface designs. Also, researchers should examine the possibility of using the HCI Context-aware Index with optimization algorithms to generate optimum designs directly from design specifications.

Acknowledgement

This work was supported in part by the U.S. Air Force Research Laboratory, Human Effectiveness Division. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Air Force.

References

[1] B. Shneiderman and C. Plaisant, "Designing the user interface," 4th ed. Boston: Addison-Wesley, 2005.

[2] T. Saunders, "System-of-systems engineering for air force capability development," USAF Scientific Advisory Board, Tech. Rep. SAB-TR-05-04, 2005.

[3] MIL-HDBK-759C: Human engineering design guidelines, 31 Jul 1995.

[4] E. H. Phillips, "F-35 cockpit technology," Aviation Week and Space Technology. Vol. 164, p. 59, 2006.

[5] R. W. Schvaneveldt, D. B. Beringer and J. A. Lamonica, "Priority and organization of information accessed by pilots in various phases of flight," Vol. 11, pp. 253-280, International Journal of Aviation Psychology, 2001.

[6] Guida, G., & Lamperti, G. “AMMETH: A methodology for requirements analysis of advanced human-system interfaces.” *IEEE Transactions on Systems, Man and Cybernetics--Part A: Systems and Humans, 30*(3), 298-321, 2000.

[7] S. Iglin. (2007, Aug) Graph theory toolbox posted in Matlab Central.. Available: http://www.mathworks.com/ matlabcentral/ .

[8] S. North 92007 Aug) NEATO graphics program posted at Graphviz. Available: http://www.graphviz.org .

[9] R. Albert, H. Jeong and A. Barabasi, "Diameter of the world wide web," Nature, p. 130, 1999.

[10] D. B. West, "Introduction to graph theory," 2nd ed. Upper Saddle River, NJ: Prentice-Hall, Inc., 2001.

[11] C. A. Phillips, “Human factors engineering,” 1st ed. New York, NY: John Wiley & Sons, Inc., 2000.

Appendix D2: Application of a Genetic Algorithm for User Interface Design

Introduction

In the modern world, most people have daily interaction with computer-based information sources. The user interfaces (UIs) that moderate those interactions are, too often, less than optimally designed. This paper extends the research on one aspect of UI design that was originally presented by the authors in [1]. In that paper, we presented an approach to display layout design based on human performance modeling. The paper proposed metrics, that, when used with the approach, allowed for a quantitative evaluation of layout effectiveness. The metrics have been positively correlated with actual human subjects experiments in [2] by performing a meta-analysis of aircraft cockpit design projects.

This paper focuses on how to use those same metrics to generate optimal layout predictions early in system design. We do this using a hybrid algorithm that generates new display layouts with minimized control operation times. The hybrid algorithm is comprised of a seeded genetic algorithm in conjunction with a pattern search algorithm.

Background

Most modern aircraft use multi-function displays (MFDs), such as the one illustrated in Fig. 1, with bezel buttons around the edges. These are called "soft buttons" as their functions can be tailored to the particular screen being displayed. This approach is now prevalent in other forms of transportation (such as automobile dashboards and boat navigation/radar systems) and public kiosks (such as automatic teller machines and airport check in stations).

Design Challenge

Sound tools for UI design, as a part of human-computer interaction (HCI), has been the

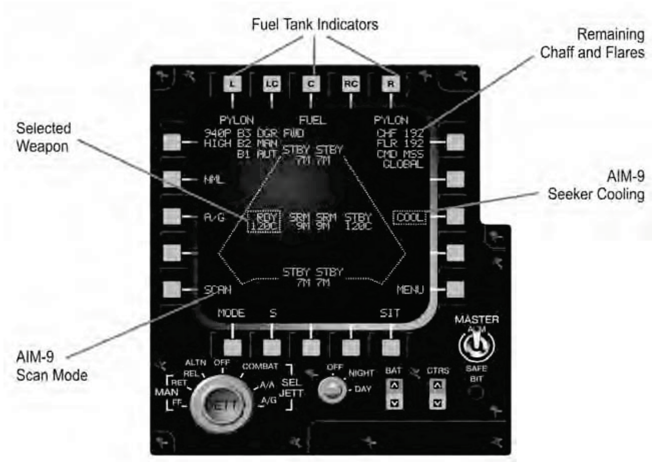


Figure 1. F-15 Programmable armament control system. Example of the use of "soft buttons" on a multi-function display. (U.s. Gov't figure)

focus of a large amount of research. Though actual user testing is regarded as the gold standard of evaluation methods, the time and expense of user testing make it prohibitively difficult to be used exclusively. Practitioners generally use qualitative inspection evaluation methods, such as expert surveys or cognitive walkthroughs in early system development [3]. These use accepted design guidelines, drawn from empirical studies, to examine the ordering, layout, and grouping of the proposed display layouts. These guidelines provide useful heuristics, but they are too general in scope to fully guide the mapping of information and functions to specific operational threads [4], [5]. Furthermore, as a whole, they often yield contradicting advice with no established structure for prioritization. Our proposed metrics quantify the established guidelines on menu structure so that they can be used to objectively evaluate displays and provide a objective function for optimization routines. The guidelines related to the design of specific displays (e.g., button order and style, the data representation, and the form of data integration) must still be accomplished.

Model Construction

A display layout can be modeled as an irreducible, nonabsorbing, time homogenous, Markov chain. A Markov chain is usually modeled as a state diagram. For display layout problems, the *nodes* (or vertices) of such a diagram are represented by v_x and represent a unique combination of information, selectable functions, and submenu options to be included in the system. We will refer to each specific combination as a *page* when the context is clear as it is consistent with website terminology and makes the analysis more readable. The *edges* (or arrows) of the graph represent transitions and are defined as node pairs $e_{a,b} = [v_a\ v_b]$. Specifically, each edge is an explicitly defined input from the human that causes a change of display state. For more on how to identify the necessary model data see [1].

Mathematically, a graph can be expressed in several matrix forms. The most intuitive is an *adjacency matrix* of binary numbers. For each i,j element of the matrix equal to one, there exists an edge between node i and node j [6]. This paper uses the term *HCI-defined G* to refer to a graph G that has been built from the definitions stated here and is thus a representative model of a valid interface design layout. By design, it will be a connected asymmetric directed graph with a specification-derived node set. The value of doing the modeling effort in this way is that numerical results can be calculated by numerous mature software packages. For our work we used the Matlab® optimization toolbox and the graph theory toolbox [7]. The graphics were generated by the Matlab® Biograph.

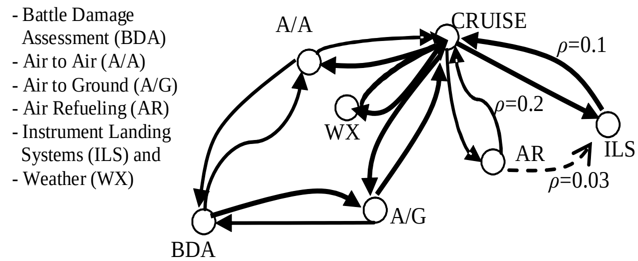


Figure 2. A sample display layout model. The nodes reflect pages for mission tasks. Sample values for $\rho_{a,b}$ are given.

Each edge has a probability of transition $\rho_{a,b}$ that is the likelihood of accessing node v_a from node v_b. For example, Fig. 2 depicts a simple graph of displays labeled with their primary task information. In it, $\rho_{\mathrm{AR,ILS}}$=0.03 < $\rho_{\mathrm{CRUISE,ILS}}$=0.20 which quantifies the fact that it is operationally less likely to transition directly to a precision approach after air refueling than it is from cruise flight. A matrix of these probabilities is called the transition probability matrix, or simply transition matrix. For a layout of *n* pages, the transition matrix is an *n* x *n* matrix of $\rho_{a,b}$ values. For existing systems the transition matrix can be formed empirically. For predictive purposes the transition matrix can be formed as follows: first, decompose the system's intended purpose into task sequences, known as operational threads. Cognitive systems engineers have produced a large amount of research on how to best do this. The recommended methods include task analysis and cognitive work analysis [3]. Secondly, total the count of each specific page change during the execution of all identified tasks. Divide each element by the sum of its row. Now the entry of row *a* column *b* is the expected $\rho_{a,b}$. Transition matrices may also be formed through the use of simulations.

Our display design method involves an edge weighting using a special form of the transition matrix. It is referred to in literature as the affinity matrix and represented by ***P*** (capital rho). To form ***P***, we tally all transitions during operational threads as described above. The affinity matrix is the joint probability mass function of the tallied matrix. By definition, the sum of all entries in ***P*** must equal 1.0. As shown in Fig. 3, in general, the transition matrix will not be symmetric. Values on the diagonal represent the number of selectable functions on a page. These are modeled as self loops as they do not invoke a page transition.

Design Analysis

Once the proposed layout has been defined as above we are able to find potential solutions. The most valuable contribution of model-based evaluation is the ability to quantitatively score proposed layouts, even before initial prototyping. Wiener, et al, proposed a sum of the shortest path distances between all nodes in a graph, now called the Wiener Index, which provides a simple goodness estimate for Markov chain analysis [8]. Liu proposed that a similar type of scoring should be used to evaluate the search efficiency of menu layouts [9]. In [1] we proposed a novel metric to do this which we

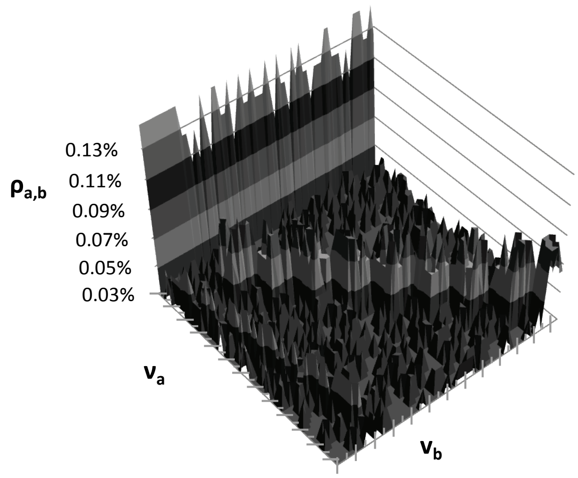


Figure 3. Surface plot of affinity matrix for 63 page example.

termed the HCI Index. This index is the expected mean control operation time of a proposed display layout. To obtain this we use a weighted shortest path calculation based on the Floyd and Warshall algorithm, as coded by Iglin [7]. The paths are weighted by mean system processing speed and expected human choice reaction time; the latter being modeled using the HCI work of Hick, Hyman, Deninger, and Wickens. The "distance" between any two edges, then, is defined as:

$$d_w(v_0, v_k) = \sum_{i=1}^{k}\left(t_i + \left(0.212 + (0.153)\log_2\left(d^+(v_{i-1}) + 1\right)\right)\right) \tag{1}$$

where:

v_0 and v_k are any two arbitrary nodes,

t_i is a time delay constant associated with the i^{th} edge on the (v_0, v_k) minimum path, and

$d^+(v_{i-1})$ is the number of selectable options of the tail node of the i^{th} edge on the (v_0,v_k) minimum path.

This metric is written as $d_{w\_G}(v_0,v_k)$ when the associated graph G is not clear from the context.

The weighted distance calculation in (1) can be used to derive a metric for evaluating the entire layout with respect to task transition time, which is also called control operation time. We write this as:

$$D_w(G) = \frac{1}{n^2}\sum_{v_a, v_b \in V(G)} d_w(v_a, v_b) \tag{2}$$

This is useful for interface design, but minimizing (2) does not guarantee that a layout is optimized for its mission, or even any specific task. For that, the metric must include input from its expected operational context. Designers can provide this in the form of the affinity matrix described above. This weights the transitions according to those most

likely to be accessed in the projected use of the system. We named this the HCI Index and write it as:

$$D_{w.\rho}(G) = \sum_{v_a, v_b \in V(G)} d_w(v_a, v_b) \cdot \rho_{a,b} \tag{3}$$

In [1] we proved that, given a set of proposed layouts, the one with the smallest value of (3) is guaranteed to have the lowest expected control operation time.

Optimal Design Problem Formulation

We are now ready to define the display layout design as an optimization problem. Other optimization approaches begin with an assumed fixed hierarchical structure and then seek to optimize the node placement in that structure [10]. This optimization effort differs in that no structure is assumed a priori and the optimization performs its selection on the set of all possible edges. Assume that designers have determined a proper node set of an HCI-defined G, an operationally-relevant affinity matrix, and a list of time constraints on time critical pages. The following paragraphs express the design problem in optimization terms.

Design Variables

Design variables are the things that can be controlled. In a menu design problem, as we have defined it, these are the edges $e_{a,b} = [v_a\ v_b]$ for all a and b from 1 to n.

Objective Function

For this problem, the objective function is the expected mean control operation time. The algorithm seeks to minimize *(3),* the HCI Index. Since the HCI Index is an implicit function, it cannot be explicitly expressed in terms of the design variables, we adapt it for use in optimization routines by coding it as a software routine which receives the design variables as inputs and returns the Index value.

Constraints

Any real world design solution will be subject to constraints. The solution must be a connected graph (no isolated components) with no stubs (a node with no out edges). Also, any real world problem will have a maximum on the number of programmed transitions from any one page. The solution must also meet constraints on the required accessibility of critical functions. These are modeled by assigning limits to the respective node maximum distance. In graph theory, the *eccentricity* of a node v_a is the maximum of the distances between v_a and all other nodes of the graph. Using (1), the *weighted in-eccentricity* is the maximum $d_w(v_b,v_a)$ for any v_b of G:

$$ecc_w^-(v_a) = \max_{v_b \in V(G)} \{d_w(v_b, v_a)\} \tag{4}$$

Simply stated, this is a measure of the greatest separation, in time, that exists between a particular screen and any part of the rest of the graph. The weighted in-eccentricity of critical nodes v_{crit} must be no greater than some maximum access time b_i [s]. Thus the

objective function is constrained by:

$$s.t. \quad ecc_{w}^{-}((v_{crit})_i) \le b_i \quad i: 1 \text{ to } m$$

where m is the number of constrained nodes.

Since this is what developers truly want to control for critical information, is a better way to identify those constraints than traditional methods.

Problem Characterization

Existence, uniqueness, and convexity

The display layout problem is explored in detail in [2]. Here, we assert without proof, that the existence of an optimal solution is guaranteed except in the overly constrained form, there is no guarantee on solution uniqueness, and the problem is non-convex.

Computational Complexity

For space reasons we also assert without proof, that the solution space is of size: $2^{n^2} - 2(n-1)$. This means that even modest problems have a very large feasible set. For example, a problem involving n=63 nodes, such as the example for a multi-role aircraft discussed in [1], has more than $10^{1,000}$ feasible designs. This quickly eliminates methods of exhaustive search as viable approaches.

Matching to the Proper Tool

In addition to being complex, both the objective function and the constraint functions are nonlinear. That is, neither the property of additivity nor homogeneity is satisfied. Also, the design variables are discrete. This precludes the use of traditional optimality methods and even numerical search methods (e.g., integer programming and sequential linearization methods) which require gradients or gradient estimations [11]. This type of specialized problem can be effectively addressed by heuristic methods such as simulated annealing, genetic algorithms, or tabu search.

We evaluated these optimization techniques and determined that this problem is best accommodated by the attributes of the genetic algorithm (GA). This meta-heuristic is based on concepts of genetic theory. An entire population of potential solutions is characterized by design variables in the same way that chromosomes characterize an organism. The GA allows us to perform global parallel searches in a highly discontinuous solution space and resist premature convergence to a local minimum [12]. Also, we are able to tune the initialization and selection criteria to exploit a priori knowledge of partial solution clustering. The work by Matsui and Yamada confirmed the superior performance of a properly tuned GA for this type of problem [13].

To implement a GA, one begins with a population of complete designs. Then:

- Test the fitness of each member using the fitness function.

- Form new generations: Select from competing members for reproduction based on fitness values. Form new designs for the next generation by performing crossover and mutation.
- Perform migration: choose some from similar groupings to move to a dissimilar grouping.
- Repeat from Step 1 until the stopping criteria is satisfied [11].

This competition within the population has enabled an aggregation of good solution subsets that were not deterministically obvious. A genetic algorithm must be designed, or tuned, for a particular problem. We have done the following tailored parameterizations.

Algorithm Tuning

Design Variable Representation

A proper representation maps the state space of the variables used in the GA population to the solution space of the actual problem. The most intuitive representation in this problem is a *n*x*n* length string of binary numbers representing the elements of an adjacency matrix, but there are numerous other graphs representations in use such as: the node list, an edge matrix, and a star representation array [14]. All were examined to see which contributed to improved computational efficiency. Michalewicz and Fogel have identified key qualities of a good representation for GA problems. They state that a representation should:

- Be a bijective mapping (that is, one to one and onto) to the real world variable set
- Maintain the link of parents to offspring; usually through a graduated range
- Be segment-able in that components of the whole are independently useful
- Always, or at least frequently, produce feasible solutions [11]

These are important for the performance of the algorithm. As it can be seen, the adjacency matrix only possesses the first quality. In fact, none of the traditional mathematical forms provide all the desired qualities.

In the pursuit of a suitable chromosomal form, we represent the design variables of the display layout by an *n*-length vector of real numbers between 1 and the edge count constraint, as defined in the specifications. Each of these values in the sequence represent the number of edges emanating from the corresponding node. The connecting node list is determined by using the affinity matrix as a look up table. For each node, the corresponding row of the affinity matrix is rank ordered by transition probability score. Edges are then assigned up to the stated number. This is not a natural representation, but it is superior to all traditional graph representations in that it possesses all of the desired qualities of a GA population. Furthermore, it uses contextual knowledge to guide the connectivity of the graph toward likely quality designs. The limitation on this representation is that it is dependent on an accurate task analysis to produce quality transition probabilities. To make this process more robust, an additional variable is

added to make it an ($n+1$)-length vector. This variable determines the amount of random noise introduced to the rank ordering of the affinity matrix elements. Thus, the algorithm is presented with a sampling from across the entire range of feasible solutions.

Since this is a custom population type, we had to modify the Matlab® creation function, mutation function, and crossover function. In addition, we had to create transformational routines that convert between the vector representation and the adjacency matrix form used by the shortest path algorithm. During this conversion we round the real value outputs of the GA into integer solution values.

Initial Population Creation

Heuristic methods are not very computationally efficient. As studied by [12], the genetic algorithm will perform better if "seeded" by feasible solutions with wide variation. To achieve this, we initially provided the GA with graphs generated using Prim's algorithm. The algorithm guarantees minimum spanning trees [15]. We found that this method did not produce an initial population with enough variation to prevent premature convergence to a local minimum. We instead provide the genetic algorithm with an initial population of pseudo-randomly generated spanning trees.

In addition, to further counteract premature convergence, two sub-populations are simultaneously evolved with full independence except at periodic migration points. Every 20 generations a mutual migration of 10% of the population occurs. These steps have successfully made the algorithm more resilient to the trap of sub-optimal local minima.

Fitness Function and Constraint Handling

The fitness function determines the solution space landscape. A good fitness function should yield some sort of correlation to the quality of the solution, that is, a score should be proportional to the distance from the optimal solution. The objective function described in (*3*) was found to do this without modification.

A primary difficulty of implementing a GA is constraint handling. Our problem requires several added procedures to account for all of the constraints discussed above. First, all variables are controlled to be between 1.0 and the edge count constraint. This means that every solution will yield at least a connected graph. Secondly, critical node constraints are evaluated as part of the scoring algorithm. Infeasible solutions are addressed by allowing an adaptive penalty function where the size of the penalty is a function of how frequently the algorithm ventures into the infeasible space. This allows transitions through infeasible space to find other optima; especially early in the search time. The penalty grows linearly as the number of infeasible solutions increase in order to force the population back into the feasible domain.

Selection Procedure

The selection procedure determines how a member's fitness function score is used to determine both its survival and chance to spawn future generations. We use the roulette technique. This is analogous to assigning members an area on a roulette wheel that is

proportional to their fitness score. This works without scaling because our fitness function algorithm returns a bounded score for all feasible solutions. In addition, we implemented an *elitist operator* in the selection process. We set this at 0.1 so the highest 10% of both populations are automatically carried to the next generation. This ensures that the best fitness level is monotonically increasing with each generation.

Variation Operators

In addition to migration and spawning, other variation operators are used to evolve the populations. We will use both mutation and crossover.

Mutation is the process of making limited random changes to fit members. We implement mutation using an adaptive feasible function. This means that the changes are made with a weighting of the success of those changes in previous generations. This makes the mutation rate self-adaptive.

Crossover is a type of multi-parent reproduction. Children are reproduced by a recombination of parts from fit members. We implement it using a heuristic function that generates children nearest the parent with the highest fitness function, but in a direction opposite from the parent with the lowest fitness function. Offspring that exceed the fitness of all parents are kept. Of those that do not, 1/n of the offspring are kept.

Final Local Search

The genetic algorithm was allowed to run for n*120 generations or until there has been a <0.1% improvement for 0.1(n*120) generations. These values were determined by observation of genetic algorithms on similar problems [11].

We implemented a two-phase hybrid algorithm in that, once the GA reached a termination criterion, the results are further analyzed by a local search algorithm. There is a great deal of work on how to best combine the explorative strength of the GA with the exploitative power of various local search algorithms [12]. We choose a pattern search algorithm because it is able to search a local "topography" without the need for a derivative or Hessian. With this approach we are able to guarantee at least local optimality.

Performance Design Method

As an initial test of proposed method, we developed design specifications for two simple node display layout problem; one four and one five pages. We then formed the transition probability matrices in accordance with the described method. For purposes of comparison, we "manually" predicted an optimal display layout by connecting the nodes with the highest affinity. We then analyzed the specification data using our algorithm. Finally, we executed a computer-based exhaustive search to find the global optimum.

In the four page example, the GA found what turned out to be the global optimum in less than three generations; though of course it continued to search until the termination criteria were met. This solution turned out to be better than the one proposed by manual design by using one less edge.

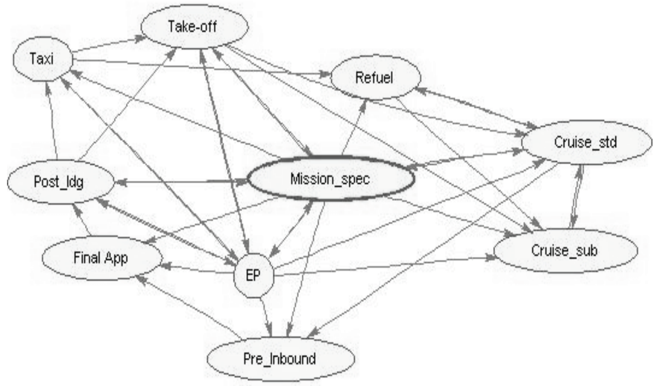


Figure 4. A ten page problem. Manual design result: 673. The GA design layout contained multiple additional edges and scored 618.

In the five page example, the GA was consistently able to find the global optimum in less than five generations. This solution turned out to be better than the implied optimal by using one less edge. This solution turned out to be better than the one proposed by manual design by using multiple additional edges. Note that even these seemingly simple problems required a computer to run several hours to perform the exhaustive search. The results are the first two sets of data points on Fig. 7.

To study the performance of the process with more complex cases, we first developed a 10 page MFD example. The algorithm was able to find a solution, shown in Fig. 4, that was 9.0% better than the one proposed by manual design.

We then revisited the 63 node problem that was originally presented in [1]. In that paper, three rational yet ad-hoc approaches were used to configure the pages of a fighter aircraft MFD. All three approaches have been used by designers of real world products. Figure 3 is a surface plot of the affinity matrix used in this problem. In [1], we scored all of the approaches with the HCI Index. The best design is shown in Fig. 5. We then analyzed the same specification data using our seeded hybrid GA. It produced a design

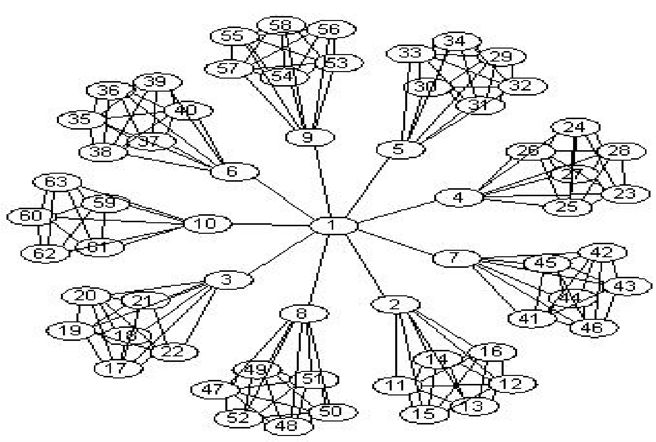


Figure 5. The best ad-hoc layout

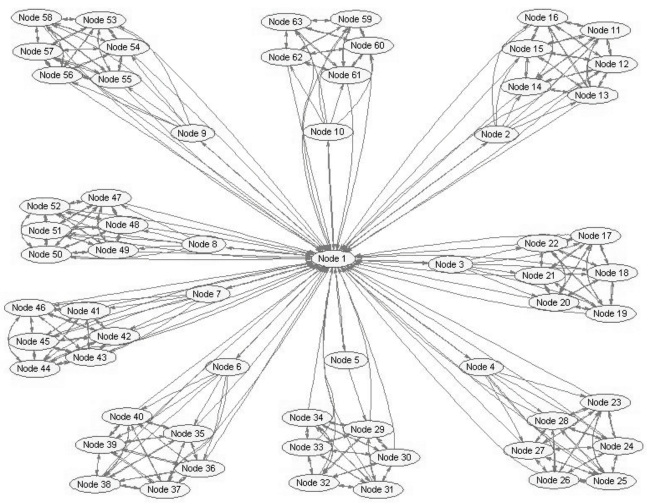


Figure 6. The hybrid GA result

that achieved an 18.0% better performance prediction. A graph of the resultant solution is shown in Fig. 6. All of the results are shown in Fig. 7. In addition, two results obtained from actual human subjects tests on displays of 41 and 43 pages are shown.

Conclusion

As it has been shown, the seeded hybrid genetic algorithm with human performance modeling is very useful for early user interface design. We have shown the necessary adaptations so that genetic algorithms can solve this problem type and even take advantage of problem-specific information. We can rapidly predict optimum designs directly from design specifications to facilitate rapid display layout development. This is widely applicable to cockpit, control station, public kiosk, and electronic device design.

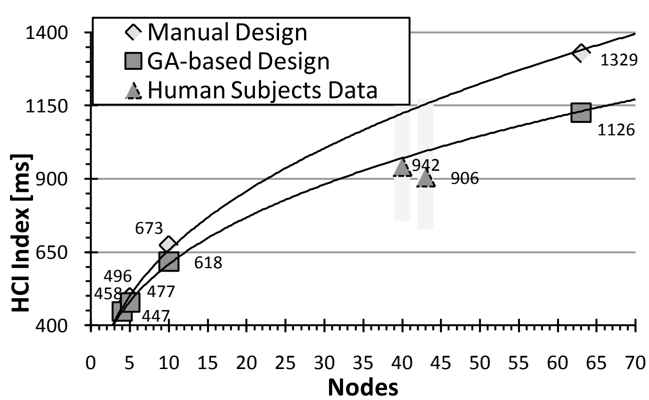


Figure 7. Comparison of manual vs. hybrid-GA design

Acknowledgment/Disclaimer

This work was supported in part by the 711th Human Performance Wing, Human Effectiveness Division.

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Air Force.

References

[1] N. Hardman, J. Colombi, D. Jacques, and R. Hill. 2008. Improved user interface design through graph-theoretic modeling. *Proceedings of IEEE International Conference on Distributed Human-Machine Systems*.

[2] N. Hardman, J. Colombi, D. Jacques, R. Hill, and J. Miller. 2009. The validation of user interface design by Markov chain modeling and seeded genetic algorthms. IEEE SMC Part A Transactions (submitted).

[3] A. Bisantz, C. Burns. Applications of Cognitive Work Analysis, CRC Press. 2009

[4] D. Kieras. Model-Based Evaluation. Chap. 60 of Human-computer Interaction Handbook. Taylor & Francis Group. Ed.s A. Sears & J. Jacko. 2004.

[5] B. Shneiderman and C. Plaisant, "Designing the user interface," 4th ed. Boston: Addison-Wesley, 2005.

[6] D. B. West, "Introduction to graph theory," 2nd ed. Upper Saddle River, NJ: Prentice-Hall, Inc., 2001.

[7] S. Iglin. (2007, Aug) Graph theory toolbox posted in Matlab Central.. Available: http://www.mathworks.com/ matlabcentral/ .

[8] M. Puterman. Markov Decision Processes: Discrete Stochastic dynamic Programming. Wiley & Sons, Inc. 1994.

[9] B. Liu. Applying Models of Visual Search to menu design. Int’l Journal of Human-Computer Studies. 56, 307-330. 2002.

[10] G. Francis. Aircraft multifunction display and control systems: A new quantitative human factors design model for organizing functions and display contents. USAARL Report No. 97-18. 1997.

[11] Z. Michalewixz & D. Fogel. How to Solve it: Modern Heuristics. Springer. 1998.

[12] J. Payne and M. Eppstein. A Hybrid Genetic Algorithm with Pattern Search for Finding Heavy Atoms in Protein Crystals. *Genetic and Evolutionary Computation Conference*. pp. 169-178. (2005).

[13] S. Matsui & S. Yamada. “Optimizing Hierarchical Menus by Genetic Algorithm and Simulated Annealing. *Genetic and Evolutionary Computation Conference*. 2008.

[14] R. Ahuha, T. Magnanti, J. Orlin. Network Flows: Theory, Algorithms, and Applicaitons. Prentice Hall, New Jersey, 1993.

[15] W. Kocay & D. Kreher Graphs, Algorithms, and Optimization. Chapman & Hall, CRC Press. 2000.

Appendix E: Other Topics of Interest

This appendix contains papers on additional areas of interest that were studied as part of this research effort. The information contained in this appendix was published in the following venues:

E1: INCOSE Insight, April 2008

Co-authored with: J. Colombi, D. Jacques, R. Hill, and J. Miller

E2: INCOSE Insight, April 2008

Co-authored with: J. Narkevicius and J. Winters

E3: Institute of Industrial Engineers, International Research Conference, 2008

Co-authored with: J. Colombi, D. Jacques, and J. Miller

E4: 7th Annual Conference on Systems Engineering Research (CSER), 2009

Co-authored with: J. Colombi, D. Jacques, R. Hill, and J. Miller

E5: 19th Annual International Symposium of INCOSE, 2009

Co-authored with: J. Colombi, D. Jacques, R. Hill, and J. Miller

Appendix E1: What Systems Engineers need to Know about HCI

Introduction

Systems engineers address the complex technical decision-making required in modern acquisition programs. This decision-making concerns the fabrication, operation, and maintenance of system elements throughout the system lifecycle. *A posteriori* analysis has identified that many programs still exceed their projected lifecycle costs or fail to achieve full utility due to the "failure to design systems that effectively integrate with human capabilities and limitations" (Booher 2003). Human systems integration is the important aspect of systems engineering that brings human issues into the system design and seeks to properly "fabricate" and "maintain" the human elements of the system. The performance of the humans in the system is built on the foundation of human systems integration in system development, as portrayed in figure 1.

In achieving human integration within systems, the relational aspect of near-ubiquitous computers and humans has become a critical performance factor. Malone and Carson (2003) express the goals of this new paradigm as "to develop a system where human and machine synergistically and interactively cooperate to conduct the mission." They state that the "low hanging fruit" of performance improvement lies in the human–machine interface block. It is essential, then, that systems engineers are equipped with the following:

1. an appreciation of the importance of human–computer interaction,
2. an understanding of the human– computer interaction community of practice,
3. an ability to evaluate the quality of a system's human–computer interaction, and
4. an ability to incorporate human– computer interaction principles into the system engineering process.

1. The Importance of the Human–Computer Interface

Maier argues that systems engineers must attend to the system's interfaces. He offers

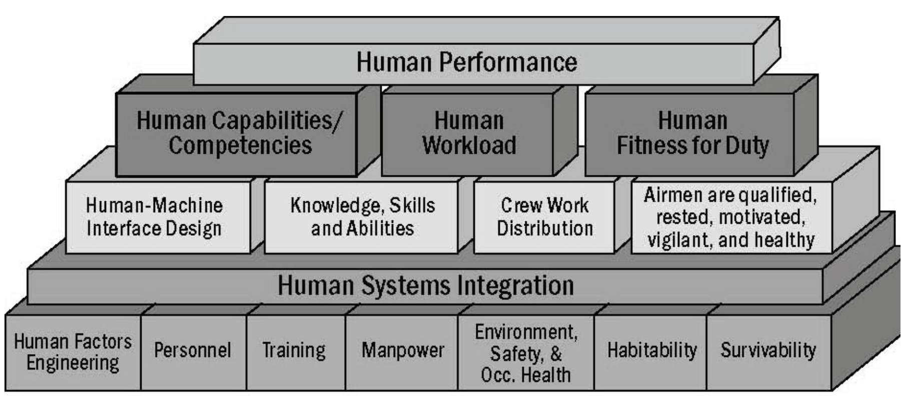


Figure 1. The domains of human systems integration (U.S. Air Force, 2005)

the following heuristic: “The greatest leverage in system architecting is at the interfaces. The greatest dangers are also at the interfaces” (Maier 1999). This heuristic is affirmed by evidence cited in the text *Designing for the User Interface*, which illuminates the high stakes of this facet of design (Shneiderman and Plaisant 2005). The “stakes” for the commercial sector are market share. To the military sector, they are tactical advantage. The *Defense Acquisition Guidebook* affirms this and identifies interface management, including the user interface, as a critical process of the systems engineer. United States Department of Defense cost studies have identified manpower, personnel, and training as comprising forty to sixty percent of the total system’s life-cycle cost (Haskins 2007). Dray states that there is a direct tradeoff between these manpower, personnel, and training costs and investment in the user interface. She cites a company project in which an improved user interface on a large-scale internal application resulted in a 32% overall rate of return stemming from a 35% reduction in training and a 30% reduction in supervisory time (Dray 1995). This result dramatically underscores the importance of the human–computer interface.

The Human–Computer Interface Community of Practice

The central emphasis of human–computer interface practice is the design of effective user interfaces; that is, the multi-modal interface between a human being and hardware. These interfaces facilitate interaction between human cognition and software logic. Researchers and practitioners have been seeking better ways to facilitate the interaction between human cognition and technology systems for decades. This work has led to the development and evolution of cognitive systems engineering. Militello, Dominguez, Lintern, and Klein (forthcoming) define cognitive systems engineering as “a design approach aimed at improving cognitive work by linking system features to the cognitive processes they need to support.” The design and development of effective user interfaces is a key activity of this community as well.

Evaluating the Human–Computer Interface

Systems engineers need to appreciate that the field of human–computer interface design can do many things for them--three in particular. The first is avoiding sub-optimal implementation that results from user rejection. The second is to reduce lifecycle costs through savings in manpower, personnel, and training. The third is to realize otherwise unattainable system performance improvements. Human–computer interface measures are less straightforward than other system requirements. Human–computer interface experts call the measure of a user interface its *usability*. Various sources expand on the concept of usability in different ways, but most group the measures similarly to the following five components (Wickens et al. 2004):

Reliability: the frequency of errors, the prevention of catastrophic errors, and the ability to recover from errors.

Efficiency: the level of productivity that can be achieved once learning has occurred.

Learnability: the amount of training necessary before the user can be productive.

Memorability: the ability for an infrequent user to maintain proficiency.

Satisfaction: the subjective experience of the user.

These last three components are unique to the user interface. Since humans cannot simply download the interface protocols into their brain, metrics such as interface learning time and ability to recall are important. The set of metrics given here is attractive because, with the use of the proper benchmarks and surveys, these metrics can be standardized and quantified (Shneiderman and Plaisant 2005). These measures aid systems engineers who must prioritize and weigh system requirements. For example, a system designed for use in combat would consider the reliability and efficiency aspects of usability as critical, but may not put a premium on user satisfaction. For activities that require an ultra-low error rate (e.g., doing something that will strongly impact others and that cannot be undone, such as firing a Tomahawk missile), the system design will emphasize error prevention (reliability) with the possible consequence of suboptimal efficiency. Across the total system requirements, these usability metrics will have to be weighed against competing requirements such as system cost and performance. These trade studies are the task of systems engineers working with a team that includes cognitive engineers and human–computer interface specialists.

Usability Metrics

Both *ISO 9341* and standard human–computer interaction texts, such as the *Berkshire Encyclopedia of HCI* (Bainbridge 2004), address how to quantitatively measure the five components of usability. Some applicable methods for systems engineering are as follows: **Reliability:** Measure the number of minor and catastrophic errors made by users while performing specified tasks. Measure the time spent recovering and fixing errors. Measure the task success rate. Note recoverability from errors. **Efficiency:** Use median users (perpetual intermediates) and measure time to perform typical tasks. Measure the number of steps to complete particular tasks. Measure the number of required workarounds for task completion. **Learnability:** Measure training time required to move

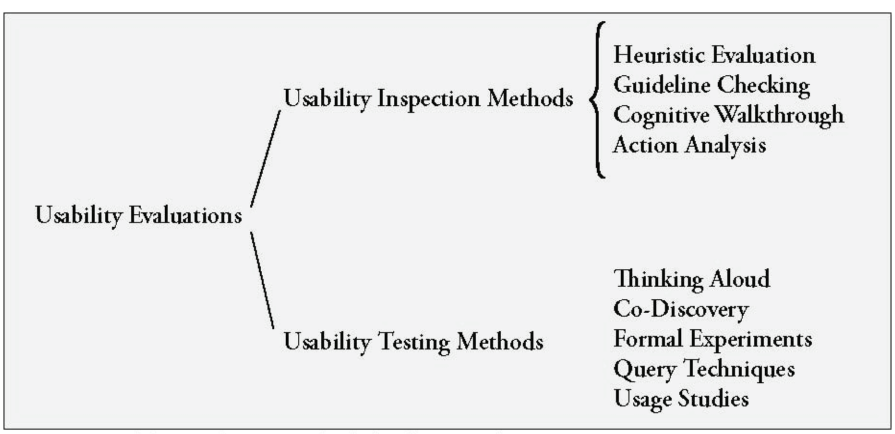


Figure 2. Usability evaluation methods (Andrew 2007)

from novice to intermediate user. Measure the frequency of help and documentation use. Memorability: Test the change in performance (speed and/or error rate) on typical tasks with frequency of use. Measure the increase in frequency of help and documentation use. Measure the number of misleading interface events. Satisfaction: Use questionnaires to discover what users found unpleasant. Measure the user's regressive behavior (when a user goes back to the old way after learning a new system). Note which features are not used. Figure 2 shows some of the techniques developed for usability evaluations. For a further discussion of these techniques, refer to textbooks such as Human–Computer Interaction (Dix et al. 2004).

Usability Heuristics

Empirical measures provide valuable and tangible insight, but systems engineers, as much as anyone, understand that not everything can be captured by quantitative means. Human–computer interaction practitioners evaluate the proper application of high-level design principles using usability heuristics. Jakob Nielson first proposed a list of ten heuristics in 1994. Though there has been no formal consensus, the list has gained general agreement in the human–computer interaction community with only mild modification. A paraphrased version is listed below with added clarification for the systems engineer (Nielsen and Mack 1994).

Feedback. The computer should continuously provide the user with visibility of the system status. For example, if a task is still being performed, a progress indicator should indicate amount of task completion. When tasks are completed, the system should indicate closure.

Real-World Agreement. System communication should be in the users' language rather than computer-oriented terms. It should also ensure that nothing misleads the users' mental model and that information appears in logical order. Metaphors must be consistent.

User Control. Users should be provided the option to undo and redo for all processes within the system's control.

Consistency. Users should be able to depend on platform conventions. If all else fails, standardize.

Error Prevention. The computer should avoid the conditions that are ripe for errors. It should require confirmation before dangerous actions (such as one that destructs or jettisons) and prompt when a probable action has been omitted (such as Save before exiting).

Recognition and Recovery from Errors. Error messages should be expressed in plain language, precisely indicate the problem, and suggest a solution and/or link to the help system.

Recognition over Recall. This is called "knowledge in the world" instead of "knowledge in the head." Whenever possible the computer should offer choices, defaults,

or example entries. If a user must enter new data, the computer should recall it, when needed, later in the dialog box.

Flexibility and Efficiency. Design features called “accelerators” are unobtrusive but offer faster interaction. This keeps the interaction simple for the novice and facilitates better performance for the expert. The computer should also allow for customization and macro-building for frequent tasks. This way the computer can cater to a greater spectrum of users.

Minimalist Design. Extra information competes with the basic information set and diminishes the relative visibility. However, all necessary information should be present. Supplementary information should be easily retrievable. Simplify tasks by providing sequential guidance or automate to relieve user memory load.

Help and Documentation. Additional information should be easy to search, focused on the present context, and list concrete steps to be carried out.

Designing Effective Human–Computer Interfaces

Developing systems that have good usability must be done deliberately. Hoffman and Elm (2006) chastise the traditional systems engineering design models (whether waterfall or spiral) as being inevitably designer-centered and urge a more *user-centered design:* a design process that establishes the user as squarely inside the system and judiciously involves future users early and often and studies their abilities, needs, contexts, and goals.

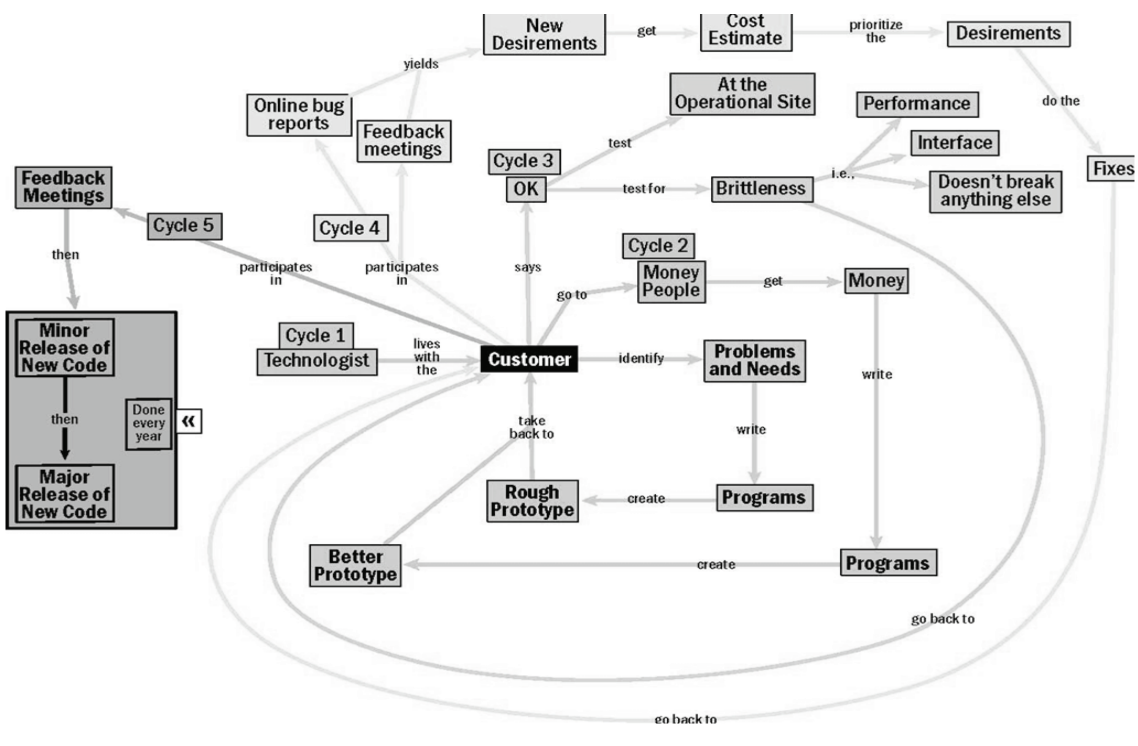


Figure 3. A practitioner’s spiral (Hoffman 2005)

A user-centered design view is shown in figure 3.

Human–computer interaction researchers advocate a usability engineering lifecycle similar to what is presented by Bias and Mayhew (2005). The primary concepts are shown in figure 4. These concepts complement good systems engineering, but they suggest some additional user-centric emphasis.

Systems engineers can address these elements in the following ways:

1. ***Know the User.*** This is part of the concept development stage. Systems engineers are well trained in requirements elicitation by identifying the user's needs, but are we failing to fully identify the user? Human–computer interaction practitioners recommend the creation of user profiles for all classes of user. This includes manpower and personnel data as well as training, operational currency, attitudes and goals, and demographic information. Some of this data must be gathered through qualitative research like observations and interviews.

What interaction patterns, which characterize the mode of operation of the human–computer interaction, should be supported? Hoffman and Elm (2006) would argue that the only way to avoid the pitfalls derived from designer-centered design is to use cognitive task analysis to understand the full set of interactions within the system. Human–computer interaction practitioners would also recommend that use cases (software descriptions of system behavior) be more specific for user type and the task context. Some advocate the use of scenario-based design, which captures more of the user attributes and interactions than traditional software engineering use cases (Bainbridge 2004). In this method, the design artifacts are narratives (similar to

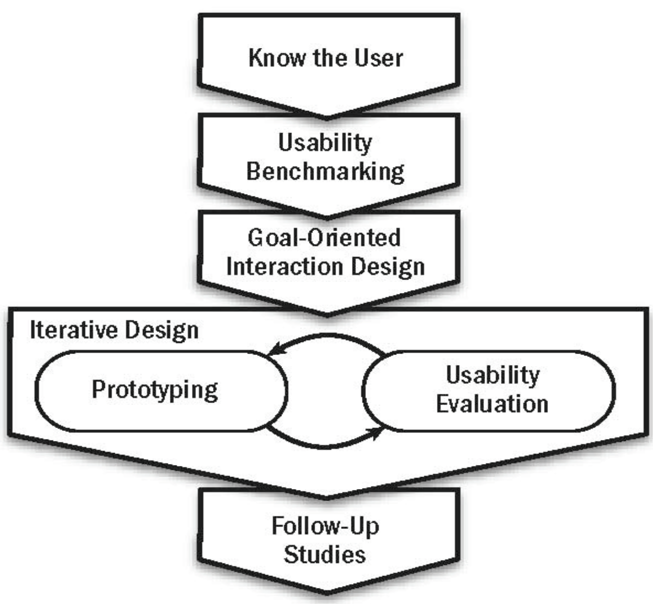


Figure 4. Usability engineering concepts (Andrew 2007)

sequences of actions found in some United States Department of Defense operational concepts) which are meant to convey a fuller sense of the actual usage conditions to the developers. It also makes greater user involvement possible because it is expressed in language that the users understand.

2. Usability Benchmarking Systems engineers must be able to incorporate usability into tradeoff analyses. This is accomplished by making measurable usability targets and comparing alternatives. New prototypes should be compared to benchmarks set with respect to an alternative option, a previous version, or a non-integrated solution. The section above on evaluating human–computer interfaces discussed how to evaluate usability both empirically and heuristically.

3. Interaction Design A designer should study the component interactions to determine what form of human–computer interaction is needed. The design team should define the system as it relates to its behavior or dialogue with other systems, people, and surrounding environment. They also must be cognizant of how system use will affect human relationships and understanding amongst the system operators (Cooper and Reimann 2006).

4. ***Iterative Design*** Systems engineers are already advocates of iterative processes. It should be emphasized, however, that we must manage the intelligent use of iterative prototypes to allow for the discovery, and subsequent correction, of usability problems as system development progresses. It is also necessary to retain design change traceability so that modification for usability issues can be explained.

Iterative prototyping should include usability evaluations both by inspection (e.g., heuristics and judgment) and usability testing (e.g., empirical testing of interface design with real users). Figure 2 showed some of the techniques developed for usability evaluations.

5. ***Follow-Up Studies*** In the support phase of the system lifecycle, important usability data should be gathered. As systems experience new releases and modifications, the follow-up studies should be ongoing and continuously capturing data. These data can be collected in a variety of ways from specific field studies (interviews, questionnaires, observation) to just tracking tech support calls or error/ deficiency reports. The important aspect for systems engineers is to put a system in place that can correctly obtain that data for use to drive upgrades.

Human–Computer Interaction: A Necessary Component of Systems Engineering

The field of human–computer interaction has the potential to increase the trade space for the total system requirements. For example, a more intuitive interface reduces demand on manpower, personnel, and training requirements, and a lower error rate improves efficiency, which reduces task execution times. Systems engineers who stay abreast of the latest human–computer interaction developments are better equipped to find revolutionary solutions to otherwise intractable tradeoff challenges. The famous technologist Don Norman says it this way (Norman 2007, 15): “We need augmentation, not automation. Facilitation, not intelligence. It is time for the science of natural

interaction between people and machines, an interaction very different than what we have today."

References

Andrew, K. 2007. Human–computer interaction: Lecture notes. Graz, Austria: Graz University of Technology. http://courses.iicm.edu/hci/hci.pdf.

Bainbridge, W. S. 2004. Berkshire encyclopedia of human–computer interaction. Great Barrington, MA: Berkshire Pub. Group.

Booher, H., ed. 2003. Handbook of human systems integration. New Jersey: Wiley.

Cooper, A., and R. Reimann. 2006. About Face 2.0: The essentials of interaction design. Indianapolis, IN: Wiley.

Defense Acquisition University. 2006. Defense Acquisition Guidebook. Fort Belvoir, VA: Defense Acquisition University.

Dix, A., J. Finlay, G. Abowd, and R. Beale. 2004. Human–computer interaction. 3rd ed. Essex, England: Pearson Education Limited.

Dray, S. 1995. The importance of designing usable systems. Interactions 2 (1): 17–20.

Hewett, T. T., R. Baeker, et al. 1996. The content of human–computer interaction. In Curricula for human-computer interaction, §2.3. New York: The Association for Computing Machinery. http://sigchi.org/cdg/cdg2.html (accessed November 29, 2007).

Hoffman, R., and W. Elm. 2006. HCC implications for the procurement process. Intelligent Systems 21 (1).

Hoffman, R. 2005. A practitioner's spiral. Glen Helen Discussions on Cognitive Engineering and Systems Engineering Co-Practice. Yellow Springs, OH: Deal Corp.

Haskins, C., ed. 2007. Systems engineering handbook: A guide for system life cycle processes and activities. Version 3.1. Rev. by K. Forsberg and

M. Krueger. Seattle: INCOSE.

Malone, T. B., and F. Carson. 2003. HSI top down requirements analysis. Naval Engineers Journal 115: 37–48.

Militello, L.G., C.O. Dominguez, G. Lintern, and

G. Klein. Forthcoming. The cognitive systems engineering landscape.

Nielson, J., and R. L. Mack, eds. 1994. Usability inspection methods. New York: Wiley.

Norman, D. A. 2007. The design of future things. Jackson, TN: Basic Books.

Shneiderman, B., and C. Plaisant. 2005. Designing the user interface. 4th ed. Boston, MA: Addison-Wesley.

U.S. Air Force. 2005. Building blocks of the airmen performance integration program. Wright-Patterson Air Force Base, OH: 311th Human Systems Wing.

Wickens, C., J. Lee, Y. Liu, and S. Gordon-Becker. 2004. An introduction to human factors engineering. 2nd ed. Upper Saddle River, NJ: Prentice Hall.

Appendix E2: Talking the Talk: Cross-Discipline Terminology Challenges

Complex systems and advanced technology are delivering good capability but continue to fall short of performance expectations when implemented in the operational setting by identified users. It has become clear that treating the system as separate from the users results in poor performance and potential failure in the operational setting. Continued growth in technology has not delivered desired results. Systems engineers and others are beginning to understand the role people play in technology systems. How will the people in the systems be represented?

The core challenge is to understand the trade space, balancing successful hardware and software solutions with human-friendly implementations. To define the requirements of humans as a fundamental system component, it is essential to understand the inherent capacity of user populations and their typical operational environment. A description of a population's capacity incorporates more than the basic anthropometrics or the cognitive capability of the average member of the user population (Booher 2003; Chapanis 1996). A population definition includes detailed descriptions and characterization of the target audience of users and maintainers, an explicit understanding of the knowledge, skills, and abilities of the people that will be operating and maintaining the system, as well as an understanding of the work that must be performed. These more diverse data must be included in systems engineering and trade space analyses, and the lack of common or consistent terminology is a barrier in this area.

Terminology issues in human systems integration are readily illustrated through the ongoing challenge of developing a unifying language for all practitioners to speak and then shaping that language to be intelligible and useful by members of the larger systems engineering community. The human domains identified internationally include human factors engineering, manpower, personnel, training, health hazards, and safety. All current human systems integration programs (termed *human factors integration* in the United Kingdom) have comparable technical practices represented, but their organization and nomenclature are not uniform. The exact number of domains and their titles is aligned with organizational structure.[1]

1. So what's the Problem? Inserting Human Systems Integration into Systems Engineering

Human systems integration is an amalgamation of pre-existing diverse technical disciplines (domains) with established vocabularies. Currently there is no single, shared vocabulary. There are words that are exclusive to a domain (making them easy to integrate into a lexicon) and there are words that are shared by many domains (making them a challenge to incorporate into a lexicon).

To complicate matters, human systems integration is part of systems engineering, itself an integration of diverse technical disciplines. For systems engineering to benefit from

the strengths of human systems integration, there must be a common language that supports the discourse and discord that allow disciplines to come together. Therefore, a lexicon for human systems integration will need to overlap with he vocabulary of systems engineering. As with the domains, the largest language barrier in communication between systems engineering and human systems integration is differing meanings for the same terms. The meaning of identical terms can differ due to the scope or level of detail associated with the term, the differences between the communities using the term or the perspective from which the term, is employed.

The challenge stems from the vocabulary itself. A vocabulary or lexicon is built on common understanding. That is, all participants know the words in the vocabulary and because of their common anchor points; they share understanding of the concepts represented by those words (Bogue 2006). Absent this understanding, programmatic and technical problems can arise, as in the following example.

I Need Effectiveness, Not Usability

Within the usability profession, *usability* is typically described as including the user's satisfaction with a product as well as the efficiency and effectiveness with which that product can be employed to achieve the user's goals. Satisfaction is typically gauged through subjective responses. Subjective methods can also be implemented to assess efficiency and effectiveness. However, efficiency and effectiveness are better measured using quantitative, objective means such as the time required to achieve a goal, error rates, or resources consumed.

But outside the usability community, those same hard, quantitative measures employed by the usability specialists are often seen as performance measures that are outside the scope of usability, leaving usability to be operationally defined as the user's subjective reactions to a product. When someone outside the usability profession has this narrowly scoped definition of usability, it becomes difficult to establish the necessary and appropriate connection between usability activities and a final product or system that performs better. Thus, differences in vocabulary can lead to misunderstanding about work scope or the connection between an area of practice such as usability and the overall development and evaluation process.

A Task By Any Other Name Still Has A Goal

Different applications of the term *task* can likewise cause confusion, especially since each meaning depends on the context provided by the higher-level meaning. For military procurements, the term *mission tasks* is used to describe the activities that a fielded unit or platform is supposed to be able to accomplish. A military unit or group of people with the same equipment may differ in the types of mission tasks they can perform due to differences in training they have received. Moving down to the level of an individual person within a unit, that individual may be assigned one or more job tasks, each of which should enable the completion of mission tasks. But a job task can only be accomplished if the individual is able to complete specific activities, which at the next level of detail may be referred to as user tasks. For a system with a substantial human-

computer interaction component, that user may have to accomplish a series of human-computer interaction tasks in the context of a goal that corresponds to a user task.

The different levels of abstraction at which the term *task* is used might not be so confusing if the relevance of each usage were not entirely dependent on the context supplied by the level above it. A human-computer interaction task is meaningless without the context of a user task, which has no purpose without a job task, which serves no useful function without a mission task. Failure to understand these levels of detail can lead to confusion and delays if an existing higher-level set of tasks is mistaken for inputs to or competition with a lower-level set, instead of being viewed as necessary context.

Alphabet Soup for Human–Computer Interaction

Human–computer interaction began as an afterthought in the programming community, but has evolved into a vast body of knowledge. Early software developers would often take the attitude represented in the motto of the 1933 World's Fair in Chicago: "Science Finds, Industry Applies, Man Conforms." Human-computer interaction combines research methods of psychology with the development processes of computer engineering. Human factors engineers, graphics designers, and experimental psychologists (Shneiderman and Plaisant 2005) have expanded it as a discipline. Though there is notable inconsistency in the terms and definitions, the authors' composite definition is that human– computer interaction is a field of study that seeks to improve all aspects of the dealings between users and computers by making computers more usable, intuitive, and accommodating of human capabilities and limitations. Thus, unlike the World's Fair motto, it is the computer that conforms.

The United States Department of Defense has listed standard terms and provided a taxonomy for topics related to human systems integration and human– computer interaction in its *Handbook on Definitions of Human Factors Terms* (*MIL-HDBK-1908B*) and its *Design Criteria Standard: Human Engineering* (*MIL-STD-1472*). Unfortunately, most service guidance uses these terms in a manner inconsistent with these publications. Several professional societies have defined their standard terminology, but none apparently has gained universal acceptance. Table 1 lists the most common terminology found across the literature and illustrates some of the confusion the variations engender.

Table E1: HCI Terminology

Acronym	Stands for	Discussion	Use
UI	User Interface	Most clear	Accepted in HCI community
HCI	Human-computer interaction Human-computer interface	When focused only on the interface, better to use "user interface"	Most widely accepted, this paper included
CHI	Computer-human interaction	Synonymous with HCI	Used in ACM and SIGCHI publications. HCI is more common
HCC	Human-centered Computing	More holistic-view than HCI	IEEE Computer Society
HSI	Human system interaction	Confused with systems engineering human systems integration	NATO, some use USAF, ISO
HFI	Human factors integration	Confused with HSI, but only the interface is often intended in HFI. In Britain, HFI is equivalent to HSI.	ISO (user-centered design)
MMI	Man-machine interface	Broader than HCI for non-computer use	Widely used in INCOSE
HMI	Human-machine interface	Avoids gender issue of MMI	
PVI, UVI, OVI	Pilot (user, operator) Vehicle Interface	Less common	USAF program offices
IxD or IaD	Interaction Design	Like HCC, represents a more encompassing discipline	Biological, social and other fields

You're doing w*hat* to the Functions?

One of the core processes supported by the human factors engineering domain is the allocation of functions to humans, hardware, software, or combinations thereof. Function allocation results drive the task analysis and interface design processes, and any function not covered by the equipment or product that is built will by default have to be accomplished by a person. Without an appropriate allocation process, the human operators may end up having to perform tasks for which they are not suited. Manpower may have to be increased by increasing the work required, or alternatively, skill requirements might increase significantly, affecting the recruiting plan. Inability to recruit the right people might result in difficulty for users who are not entirely suited to the work.

Therefore, when a human factors practitioner hears that a function allocation activity is underway, the typical reactions are to try either to get involved or to be thankful that the

foundation is being laid for future human factors work. But this is not always the correct reaction, since the human factors definition of function allocation is not always the one that is in use. Other definitions include function allocation processes taking place at a much higher level, with major functions being allocated to subsystems to be designed later or to pre-existing components that will be integrated to form the full system. Since the pre-existing components will bring with them their inherent allocations of function to human and machine, these activities are still relevant to the human factors practitioner. However, function allocation can also refer to the process of allocating functions back to the requirements which they will satisfy (ANSI 1999), which would not be an activity of much relevance to the human factors or human systems integration practitioner.

The Individual's Solution

At the individual level, there are a number of things the practitioner can do to avoid misinterpreting information due to lexicon differences. First, account for the background of the individual employing the potentially problematic terms: less overlap in background means a higher likelihood of inadvertent miscommunication. Second, do not rely upon "key words" and individual terms to gain an understanding. Not only should you pay attention to the context of the terminology, but if you do not get enough background to confirm the intended meaning, you need to ask for it. Although differing processes may be titled the same, they are not as likely to have the same inputs and outputs or work with the same subject matter if they are in fact different activities.

Similar tactics should be employed when introducing terminology. Be aware of the diverse background represented in the audience, and provide context for any potentially problematic terms. Rather than introducing entirely new terms, the easiest course of action is often to introduce modifiers for the terms. Instead of referring to a general "task analysis," labeling it as a job task analysis or human-computer interface task analysis will reduce ambiguity.

The Community Solution

While individual efforts can avoid some terminology pitfalls, it is of central importance that we develop a formally documented lexicon. Documentation of the lexicon will facilitate shared understanding, encourage people to employ less-frequently-used words that consequently result in fewer misunderstandings, and provide a platform for an ongoing discussion of meaning while allowing for change over time and usage.

A common vocabulary is predicated on common experience. While a lexicon may grow organically, a formally documented vocabulary (including definitions) ensures that all participants can participate and calibrate their usage. An example that is easy to follow is available from Retiveau, Chambers, and Esteve (2005), who developed a lexicon for describing the flavors of French cheeses. The process included identifying the attributes to fully describe the flavors, categorizing those attributes, researching published lexicons, and adopting a new lexicon.

Human systems integration can follow a similar process while including rigorous peer review. The human systems integration lexicon should

- identify comparability of terms, based on definition and data collected,
- align those terms by data equivalence, with caveats and descriptions as needed,
- instigate subject matter expert review, employing subject matter experts across domains—which may have small populations,
- develop a consensus, as language will only thrive via consensus and shared use, and
- document and publish the vocabulary and definitions in a codified lexicon, presented in an accessible and usable format.

INCOSE's Human Systems Integration Working Group has begun an effort to develop this formalized lexicon using tools such as concept mapping as a means to collect and codify the vocabularies of the human domains and of general systems engineering. This effort continues but is predicated on participation across the human systems integration/systems engineering community, and is clearly a starting point. It is essential to understand that the lexicon represents a snapshot of the vocabulary and its usage for a moment in time. It will require updating, much like the Oxford English Dictionary, to incorporate new words, modify changing definitions and usages, and to delete outmoded words no longer in use. The language problem begins to correct itself as human systems integration and systems engineering practitioners become more familiar and more comfortable working together on a regular basis.

Including the human as a fundamental system component is part of the trade space in any system design and is an essential part for advanced technology systems. The challenge, however, is to understand this portion of the trade space by balancing "human-friendly" solutions with successful hardware solutions and software implementations. Human systems integration will bring a great deal of improvement to systems engineering by ensuring that the human elements of the system are considered throughout the systems engineering process as part of the trade space. Successfully integrating human systems integration into systems engineering will require a shared understanding and a means to communicate successfully. Systems engineering is a human activity and therefore is highly dependent on communication to make proactive tradeoffs, describe all the system's elements and environment, and produce a useful output.

References

ANSI (American National Standards Institute). 1999. ANSI/EIA-632: Processes for engineering a system. Washington, DC.

Bogue, R. L. 2006. Creating a common lexicon for software development in your organization. TechRepublic (Web site). http://articles. techrepublic.com.com/5100-3513-6081740.html (accessed 25 Feb. 2008).

Booher, H. R., ed. 2003. Handbook of human systems integration. Hoboken, NJ: Wiley.

Chapanis, A. 1996. Human factors in systems engineering. Wiley Series in Systems Engineering and Management. Andrew Sage, series editor. Hoboken, NJ: Wiley.

Martin, J. N. 1997. Systems engineering guidebook: A process for developing systems and products. Boca Raton, FL: CRC Press.

Novak, J.D., and A. J. Canas. 2007. The theory underlying concept maps and how to construct them: Technical report IHMC CmapTools 2006-01 Florida Institute for Human and Machine Cognition. http:// cmap.ihmc.us/Publications/ResearchPapers/ TheoryUnderlyingConceptMaps.pdf (accessed 25 Feb. 2008).

Rétiveau, A., D.H. Chambers, and E. Esteve. 2005. Developing a lexicon for the flavor description of French cheeses. Food Quality and Preference 16 (6): 517–527. Available through http://www.sciencedirect.com (accessed 25 Feb. 2008).

Shneiderman, B., and C. Plaisant. 2005. Designing the user interface: Strategies for effective human– computer interaction. 4th ed. Reading, MA: Addison-Wesley.

U.S. Department of Defense. 2003. Department of Defense Directive 5000.1: The Defense Acquisition System. Washington, DC: Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics.

Appendix E3: HSI within the DoD Architecture Framework

1. Introduction

The need to integrate human considerations within and across all elements of system development is gaining increased attention in recent literature, but most authors admit that many projects still fall short of true integration. The human-systems integration (HSI) components of system requirements are not easily quantified and the tradeoffs are often analytically complex. The U.S. military uses the Department of Defense Architecture Framework (DoDAF) as a modeling aid to support complex system design. Architectures are the structure of various design components. They map the dependencies between system elements. This paper explores how to document and analyze the HSI components within the overall definitions of the current DoDAF (Ver. 1.5), mandated for use in August of 2007. This paper also proposes some DoDAF modifications that would enable the use of simulation to evaluate the behavioural and performance aspects of HSI-inclusive models. The authors assume that the reader has a basic familiarity with the architectural products described by DoDAF. For more information and an introduction to these products, consult [1].

DoDAF Capabilities and Deficiencies for HSI

HSI requirements and tradeoffs present problems that are not easily quantified. Architectural products are the place to address these issues, because, as Maier and Rechtin state, "Architecting handles ill-posed or ill-structured situations." [2]. The user interface, a key focus of HSI, must be architected in DoDAF to prevent the risk of system failure. User interfaces are relationships between critical components of the system and, "Architecting is the structure of components, their relationships, and the principles and guidelines that govern their design and evolution over time" [1]. For architectures using DoDAF, this structure of components is captured in the underlying logical data model called the Core Architecture Data Model (CADM). This is useful for assessing the dynamic behavior of the system, and is an essential prerequisite for building executable models to observe simulated behavior. If managed properly, these modeling and simulation efforts will yield savings in the cost and schedule of system development.

As defined by the International Council on Systems Engineering (INCOSE) Handbook, HSI is "the interdisciplinary technical and management processes for integrating human considerations within and across all system elements; an essential enabler to systems engineering practice."[3]. Figure 1 presents the nine domains of HSI using the taxonomy of the US Air Force HSI office. Systems engineers do not replace the disparate organizations of each domain, but they must effectively interact with them in order to facilitate tradeoffs and integration during system development. DoDAF has the potential to help: integrate the HSI domains, integrate HSI with other engineering disciplines, provide actionable information for the HSI products, and inform HSI tradeoff analyses

Figure 1: HSI Domains

[4]. The following sections give summarized definitions from the INCOSE SE Handbook of each domain and an analysis of the DoDAF 1.5 capabilities for the respective domain.

Manpower

The Manpower domain determines the number and type of personnel required to operate and support a system. Support includes functions such as maintenance, sustainment, and training. Many civilian organizations call this "human resources".

DoDAF capability: The "human" in HSI includes all personnel (owners, customers, operators, maintainers, support personnel, etc.) who interact with the system. An appropriately developed set of Operational and Systems/Services Views will communicate the exact roles of the human system components and their interaction with other system components. Furthermore, the DoDAF offers the incorporation of use cases into the architecture. This information is essential for the overall HSI Plan as well as the Manpower Estimate and the Training System Plan. It can also aid the systems engineer to find the most efficient use of manpower in a system by capturing the number and types of personnel needed for proposed configurations.

Personnel

The Personnel domain determines the attributes (knowledge, skills, and abilities; KSAs) and the physical, cognitive and sensory capabilities required of the humans in the system. The personnel community defines these parameters for the system and determines the how to best obtain and maintain an adequate pool of qualified persons. The U.S. Army calls it "personal capabilities" and it is inter-related with human resources in civilian organizations.

DoDAF capability: Designers should use a set of Operational Views to discover and communicate the context of the human system components. Human operator functional allocation is identified in the systems and services views. If thoroughly done, this should provide the personnel community with the attributes and capabilities that will be required of the humans in the system.

Training

The Training domain determines the necessary infrastructure and system components to provide system personnel with the requisite attributes (KSAs) for optimal system performance. This includes individual and unit training programs, training systems, and retraining schedules.

DoDAF capability: Manpower, Personnel, and Training (MPT) are highly coupled. Tradeoff analyses invariably involve the interrelations of these domains and their composite relation with human factors. As mentioned, the operational use cases, together with the Operational Node Connectivity Description (OV-2) and Systems Interface Descriptions (SV-1) should be used to gain critical information for the Training System Plan. The personnel assigned to the nodes identified in these products must be trained on all of the interactions and processes that they will be expected to perform. Likewise, the Systems Performance Parameters Matrix (SV-7) should capture non-functional performance objectives for the human system.

2.4. Human Factors

The Human Factors (HF) domain addresses how to incorporate human characteristics and limitations into system design for optimal usability. The issues of this domain are often divided into the following categories:

- Cognitive— e.g. response times, level of autonomy, cognitive workload limitations
- Physical— e.g. ergonomic control design, anthropomorphic accommodation, workload limitations
- Sensory— e.g. perceptual capabilities, such as sight, hearing or touch
- Team dynamic—e.g. communication and delegation, task sharing, crew resource management

A primary concern for HF is the creation of effective user interfaces. European organizations generically refer to human factors as "ergonomics" and much of industry calls this "human factors engineering (HFE)".

DoDAF capability: The human operators must be adequately modeled in DoDAF to ensure that function allocation meets system requirements. A fully developed set of systems and services views will capture HF task analyses, both physical and cognitive, in the delineation and allocation of functions. The Systems Interface Description (SV-1) is important because it captures all interfaces between system components (human or otherwise). Details regarding necessary interactions between human operators and computers in the system should be documented in the Systems Communication

Description (SV-2) and characteristics of the interface should be described in Systems-Systems Matrix (SV-3). For all user interfaces, this should include the cognitive and sensory limitations of the human. In addition, the DoDAF lists HF capabilities as one of the technical standards to be contained in the Integrated Dictionary (AV-2). It also identifies user interface and graphical user interface functions as some of the system functions to be documented in the Systems Functionality Description (SV-4a).

The above mentioned architectural products can be used to identify HF problem areas (such as the requirement for unavailable skills or the potential for frequent or critical errors). They can also be used to identify ideal tradeoff opportunities between Manpower, Personnel and Training (MPT) and HF. For example, the architectures can identify the costs/benefits of designating certain functions for automation. A historical example of this is the platform redesign for the suppression of enemy air defenses (SEAD) mission. Improved computer automation of the signal processing progressively allowed the mission requirements to be reduced from 4 pilots/2 aircraft (F-4 hunter/killer team) to 2 pilots/1 aircraft (F-4 upgraded system) to the current 1 pilot/1 aircraft (F-16CJ). The benefits in cost and personnel safety quickly made up for the cost to develop more complex automation.

System Safety

The System Safety domain evaluates the characteristics of systems in order to minimize the potential for accidents.

DoDAF capability: The System Safety Analysis, a required program management product, looks for design features to prevent safety hazards where possible and to manage safety hazards that cannot be avoided. The systems and services views aid this analysis by helping to identify the need for back-up systems and the incorporation of safety alerts in the user interfaces. The context of failure modes and the events (internal or external) that trigger them should be captured in the System State Transition Diagram (SV-10b). The Technical Standards Profile (TV-1) documents standards that constrain the systems of the SV-1. The human limitations, such as those identified in the HSI domains of safety, health, survivability, and habitability, belong in that repository.

Survivability

The Survivability domain evaluates the characteristics of systems that can reduce fratricide and probability of attack, as well as minimizing system damage and injury if attacked.

DoDAF capability: Survivability analyses require the evaluation of a wide spectrum of system components such as: oxygen supply, armor, egress/ejection equipment, seat belts, electronic shielding, and anti-virus protection. Furthermore, survivability issues must be considered in the full spectrum of anticipated operational environments and for all humans that will be a part of the system. A thorough concept of operations (or CONOPS) can help systems engineers cover many of these aspects. Specific data incorporated into the systems and services views can identify how survivability issues are

being addressed (e.g. required equipment for human operators performing specific functions in specified locations). As mentioned previously, the TV-1 should also contain applicable survivability standards.

Health

The Health domain evaluates the characteristics of systems that create significant risks of injury or illness to humans. Sources of health hazards include: noise, temperature, humidity, NBC (nuclear, biological, and chemical) substances, physical trauma, and electric shock.

DoDAF capability: During system development, program management teams perform several health assessments to consider the potential for harm to humans from conditions such as: operator location, operating characteristics, and surrounding equipment. To do this, they must know the manual task location and be able to identify the tradeoffs of relocation. The systems and services views aid this analysis by identifying the locations of operators in the SV-1. The system views should also capture the health warning mechanisms as part of the user interface in the System Data Exchange Matrix (SV-6) and Systems-Systems Matrix (SV-3). Also, the Operational State Transition Description (OV-6b) and Operational Event-Trace Description (OV-6c) should be used to examine the potential health and safety risks from specific event sequences. Lastly, as mentioned previously, the TV-1 should contain applicable health standards.

Habitability

The Habitability domain evaluates the characteristics of systems that have a direct impact on personnel effectiveness by maintaining high morale, comfort, and quality of life.

DoDAF capability: The system habitability of a system incorporates such wide ranging issues as the physical layout, climate control, and sanitation and service facilities. The DoDAF products useful for health and survivability analyses, as previously discussed, will be useful for habitability analyses as well. In addition, the TV-1 should contain applicable habitability standards.

Environment

The Environment domain evaluates the system in the medium for operation and the relationships which exist among all living things and systems. Consideration is made to protect the environment from system manufacturing, operations, sustainment, and disposal activities.

DoDAF capability: Environmental considerations may have a large influence on the system CONOPs. As already discussed, the CONOPS, and corresponding Operational Concept Graphic (OV-1), should be expanded to address HSI (including environmental) issues. Several templates already exist to describe CONOPS; these include Air Force Policy Document (AFPD 10-28) and IEEE 1362. The incorporation of an “environment” operational node in the OV-2 and the “environment system” in the SV-1 can be used to capture critical environmental aspects early in the design.

Suggestions for Better Human Modeling in DoDAF

Humans have complex roles, capabilities, and make decisions that have great influence on the system performance. System engineers recognize that human operators must be modeled in any complex system analysis. Though no tools exist at this time that make this fully possible, the following sections describe improvements to DoDAF that would make the concept and requirements analysis process more in agreement with human-centered design concepts. This is described by Wickens et al. as "human and machine synergistically and interactively cooperat[ing] to conduct the mission." [5].

Restore and expand DoDAF human factors instruction.

In the transition from DoDAF version 1.0 to 1.5, an entire section (Vol. I, Sec. 4.4) was removed. This section instructed designers how to address human factors within the architecture. For example, it laid out a design sequence for HCI that recommended a fully developed set of operational views prior to user interface component design. This enables designers to verify that all necessary information will be displayed for given tasks. The section also discussed how to use the logical relationships between the human and the machine in HSI requirements tradeoff analysis [6].

Provide cohesive instruction on incorporating HSI in DoDAF.

Section 2 of this paper detailed numerous ways in which the DoDAF *should* be a more effective tool for all systems engineering efforts through better inclusion of HSI concepts. However, these ideas were drawn from multiple sources and multiple sections within the DoDAF instructions. What is lacking is a cohesive guide for HSI incorporation in an integrated architecture. This would be invaluable for HSI as the interdependent domains must be considered holistically. This guide could be a new section or a separate manual and should include the type of guidance discussed in suggestion 1, but for all domains of HSI.

Clarify the HSI terminology.

The current DoDAF uses the term *human-computer interface* and abbreviates it HCI. This causes much confusion as the generally accepted term in literature is user interface and HCI is an abbreviation for the much broader concept of human computer interaction. Also, several DoD publications use both HCI and HSI terminology inconsistently. For example, the Defense Acquisition Guide has a different taxonomy for the domains of HSI than those of INCOSE and the USAF AIRPRINT as presented above [7]. This causes communication friction and means that the scope of the HSI effort varies by organization. Furthermore, the glossary needs a more complete listing of HSI-specific terms (such as the domains and effective HSI design methodologies).

Add more HSI aspects to the CONOPS template.

This recommendation was mentioned in Section 2. The CONOPS is the supporting text of the OV-1 and should contain a user description that documents the needs, goals, and characteristics of the whole user population. The primary objective of HSI is to integrate the human as a critical system component; be it as individuals, crews, teams, units, or

organizations. A fully developed CONOPS would be a valuable tool in this effort. In addition, the CONOPS should fully capture the expected environment and environmental impacts of the system's manufacture, operations, sustainment, and disposal.

Implement the ASM.

The current DoDAF recommends the architectural specification model (ASM) which is based on the activity-based methodology (ABM), as a process for creating architectural products. The ASM is a six step process that seeks to address problems identified in earlier implementations of DoDAF. One problem that still remains is that DoDAF overloads the term "system" and treats humans and man-made systems the same. While in rare instances they may act analogously, often due to cognitive abilities, they are very different. The cognitive component of the human and the software need to interact [8], facilitated by good HCI. The human cognition interfaces with the world through the sensory and anthropomorphic realm and the computer software interfaces with the world through the communications and network hardware. The ASM separates the "who" into a human performer view and an asset view. This is meant to ensure a design that fully incorporates human capabilities and limitations on the component interactions.

Conclusions

Improvements should be made to DoDAF because an architecture without HSI considerations will fail to achieve its potential utility. HSI activities are essentially systems engineering activities (requirements analyses, capability gap analyses, analyses of alternatives, task/function analyses, function allocation, and tradeoff analyses) that involve critical components, the humans. The authors recommend that the HSI efforts (such as the Manpower Estimate, Training System Plan, HSI Plan, System Safety Analysis) remain separate from DoDAF but leverage off the data "hooks" of the DoDAF products, as described above. This places the information in a standard structure, but allows the specific HSI documents to evolve with new HSI doctrine. An architecture which models the human and fully incorporates HSI requirements will provide better design information to engineers and a better decision analysis tool for program management.

Acknowledgements

This work was supported in part by the U.S. Air Force Research Laboratory, Human Effectiveness Division. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Air Force.

References

1. DoD Architecture Working Group, "Department of Defense architecture framework (DoDAF) Version 1.5," Volumes I and II, 23 April 2007.
2. M. Maier, E. Rechtin, The Art of Systems Architecting, 2nd ed., CRC Press, Boca Raton , FL, 2002.
3. INCOSE, INCOSE Systems Engineering Handbook. Ver. 3.1, 2007.

4. A. Bruseberg, “Applying the Human Views for MODAF to the conception of energy-saving work solutions,” Human Factors Integration Defence Technology Centre, Bristol, UK, 2008.
5. C. Wickens, J. Lee, Y. Liu, and S. Becker, "An Introduction to Human Factors engineering," 2nd ed. Upper Saddle River, NJ, 2004.
6. DoD Architecture Working Group, "Department of Defense Architecture Framework (DoDAF) Version 1.0", Volumes I and II, 9 February 2004.
7. Defense Acquisition University. (2006, May). Defense Acquisition Guidebook. Available: http://akss.dau.mil/dag/welcome.asp
8. H. W. Ang, D. Nicholson and B. Mercer, "Improving the practice of DoD architecting with the architecture specification model," The MITRE Corp., McLean, VA, Tech. Rep. Case #05-023, JUN, 2005.

Appendix E4: Challenges of Human Consideration in the SE Technical Processes

1. Introduction

In this paper we study acquisition oversight reports, aircraft mishap investigations, and current systems engineering (SE) literature in order to form a current picture of the most significant SE challenges. This research focuses on the SE technical processes for design as defined in U.S. Department of Defense (DoD) policy; however, the DoD has co-evolved their SE processes with the global systems engineering community. Therefore, the reader should see that these recommendations readily generalize to the process activity roadmaps of other communities such as those described in the International Council on Systems Engineering (INCOSE) SE Handbook, Institute for Electrical and Electronics Engineers (IEEE) Standard 1220, or the International Organization for Standardization (ISO) Standard 15288 (IEEE, 2008; INCOSE, 2007; ISO/IEC, 2007).

Background

Before engaging in a discussion of current systems engineering challenges, some background information is necessary.

System Lifecycle

Systems engineering encompasses the entire system lifecycle, from concept to disposal. The ISO 15288 defines the six stages of a system lifecycle as: concept, development, production, utilization, support, and retirement (IEEE, 2008). For the DoD, the revised acquisition lifecycle has been defined in DoD Instruction (DoDI) 5000.02 as shown in Figure 1. This guidance reflects the increased focus on work that occurs prior to official program initiation at Milestone B (DoD, 2008).

The Joint Capabilities Integration Development System is the first source of input for systems engineering technical processes in the DoD. It begins with a structured mission

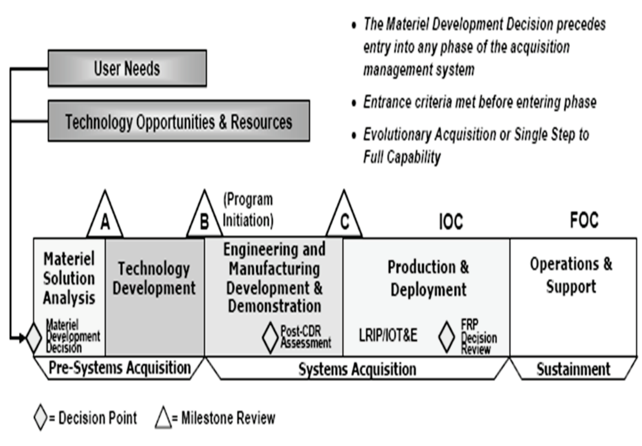


Figure 1 – DoD Acquisition Lifecycle [4]

analysis to determine how to fill identified capability gaps for the military. If the solution to a particular capability gap includes new product development efforts, the resulting analysis leads to a Materiel Development Decision, which precipitates the beginning of a system's lifecycle (JCS, 2007), as shown in Figure 1.

Human Systems Integration

Human systems integration (HSI), as defined by INCOSE, is "the interdisciplinary technical and management processes for integrating human considerations within and across all system elements; an essential enabler to systems engineering practice" (INCOSE, 2007). Systems engineers have traditionally relied on ad hoc means for incorporating human considerations in design decisions. HSI is concerned with providing methods and tools that support the SE community by ensuring humans are considered throughout the systems development process in a logical and effective way (Pew & Mavor, 2007). To date, there is no universal agreement as to the delineation of domains. As it was shown in Hardman et al, while there is general agreement over the original domains of manpower, personnel, training, and human factors, some communities have recently added domains that have not been fully embraced by the rest (Hardman, Colombi, Jacques, & Miller, 2008). Figure 2 presents the nine domains of HSI using the nomenclature of the DoD.

Systems engineers do not replace the distinct communities of each domain, but they must effectively interact with them. This is a growing segment of SE literature, but studies reveal that many projects still fall short of effectively integrating humans in the systems engineering processes (Bainbridge, 2004; Bias & Mayhew, 2005; CSE, 2008a; Dray, 1995; GAO, 2005; Harris & Muir, 2005; Malone & Carson, 2003; Mayhew, 1999; Norman, 2007).

The sections that follow discuss the HSI domains more completely. Though there is no universally accepted set of definitions, there is much in common among the definitions found in current literature. The key components have been extracted. These are written in terms that enable a system engineer to delineate system requirements by domain and to perform tradeoff analysis between domains. While these are original, they draw heavily from the definitions put forth by INCOSE, and the Human Effectiveness Directorate of the Air Force Research Laboratory (AFRL) (AIRPRINT, 2005; INCOSE, 2007).

1.1.1 Manpower

The manpower domain determines the number and type of personnel required to operate and support a system. Support includes functions such as maintenance, sustainment, and training. Many civilian organizations call this *human resources*.

DoD direction on manpower estimates for major defense acquisition programs is extensive, but there is a lack of tools to enable implementation. U.S. acquisition law requires that the Secretary of Defense consider the estimate of the personnel required before approving any new system. The DoDI 5000.02 satisfies this by requiring a formal manpower requirement estimate at Milestones B and C of the system lifecycle. Also, a

final manpower estimate is part of the full-rate production decision review. Program managers must coordinate with the manpower community, and the manpower estimates must be approved by the respective service assistant secretary. (DoD, 2008).

1.1.2 Personnel

The personnel domain determines the knowledge, skills, and abilities and the physical, cognitive and sensory capabilities required of the humans in the system. The personnel community defines these parameters for the system and determines how to best obtain and maintain an adequate pool of qualified persons. The U.S. Army calls it *personal capabilities* and it is related to *human resources* in civilian organizations.

1.1.3 Training

The training domain determines the necessary infrastructure and system components to provide system personnel with the requisite attributes for optimal system performance. This includes individual and unit training programs, training systems, and retraining schedules.

Issues within the manpower, personnel, and training domains are often inexplicably connected. This requires tradeoff analysis to be done with regard to the effects on all three. For example, there is a great deal of pressure on program managers to get early and accurate data regarding manpower requirements; however, manpower estimates for a system under development are interrelated with the methods and decisions of the personnel, training and human factors domains. This means the manpower analysis must be connected to many tangential documents in the system development process. Currently, those manpower estimates are based on heuristic comparisons with legacy

Figure 2 – HSI Domains [20]

systems. New technologies may cause these legacy-based estimates to be highly flawed.

1.1.4 Human Factors

The human factors domain addresses how to incorporate human characteristics and limitations into system design for optimal usability. A primary concern for human factors is the creation of effective user interfaces. The issues of this domain are often divided into the following categories:

Cognitive— response times, level of autonomy, cognitive workload limitations

Physical— ergonomic control design, anthropomorphic accommodation, workload limitations

Sensory— perceptual capabilities, such as sight, hearing, or tactile

Team dynamic— communication and delegation, task sharing, crew resource management

European organizations generically refer to human factors as *ergonomics* and much of industry calls this "human factors engineering". The methods and tools of this domain are the most mature of all HSI.

The DoD has produced extensive guidance on human engineering. This is a term related to the human factors domain, but used in the DoD to describe the application of anthropomorphic and physiological data for system design. The DoD has created MIL-HDBK-759C: *Human Engineering Design Guidelines* (DoD, 2003b). For example, the handbook discusses the control input distribution between hand and foot controls. Foot controls should be used only for coarse adjustments and should not be used to adjust a visual display. Hand or arm-operated controls are desirable for fine adjustment, but the more precision required, the less arm movement should be involved (DoD, 1998). These requirements are based on valid human subjects research and have the potential for a valuable contribution to SE if they can be better incorporation into methods and tools.

1.1.5 System Safety

The system safety domain evaluates the characteristics and procedures of systems in order to minimize the potential for accidents. Safety studies affect system design by advocating features that eliminate hazards when possible and manage them when they cannot be avoided. Such features include sub-systems for: system status, alert, backup, error recovery, and environmental risk.

Modern perspectives in mishap investigation follow the theories of Dr. James Reason who proposed that a hazard becomes an accident through the ill-fated alignment of certain latent conditions in the systems that are in place to prevent such accidents (Reason, 1990). The analysis performed by accident investigation boards in legacy systems can reveal valuable insight into the design of the next generation of systems.

1.1.6 Survivability

The survivability domain evaluates the characteristics and procedures of systems that can reduce the probability of attack or fratricide, as well as minimizing system damage and injury if attacked.

1.1.7 Health

The health domain evaluates the characteristics and procedures of systems that create significant risks of injury or illness to humans. Sources of health hazards include: noise, temperature, humidity, CBRNE (i.e.: chemical, biological, radiological, nuclear, and explosive) substances, physical trauma, and electric shock.

1.1.8 Habitability

The habitability domain evaluates the characteristics and procedures of systems that have a direct impact on personnel effectiveness by maintaining morale, comfort, and quality of life. These characteristics uniquely include: climate control, space layout, and support services.

1.1.9 Environment

The Environment domain evaluates the system in the medium for operation. Consideration is made to protect the environment from system manufacturing, operations, sustainment, and disposal activities.

Systems Engineering Processes

The systems engineering community addresses the challenges of systems development through prescribed processes. These processes should be followed throughout the system lifecycle and across all domains. A process is a sequence of activities performed for a given purpose (DAU, 2006). It delineates *what* needs to be done, but generally does not dictate *how* to do it. Various communities have established standards to formalized SE processes. "Standards are meant to provide an organization with a set of processes that, if done by qualified persons using appropriate tools and methods, will provide a capability to do effective and efficient engineering of systems." (DAU, 2006).

Beginning in the early 1990s, the DoD acquisition community was reformed and military standards (MIL-STDs) were eliminated. Motivated by this, the Government Electronics and Information Technology Association (GEIA) created EIA-632: *Processes for Engineering a System* from what was MIL-STD-499. GEIA has now merged with the Information Technology Association of America (ITAA) and the newest revision of the standard has been adopted by ANSI (ANSI/GEIA, 2003). This standard is a high-level look at the processes that should be standardized industry-wide. By intent, this standard must be supported by methods and tools developed for specific organizational application. The GEIA concept for the iterative relationship between formal processes in a system lifecycle is portrayed in Figure 3.

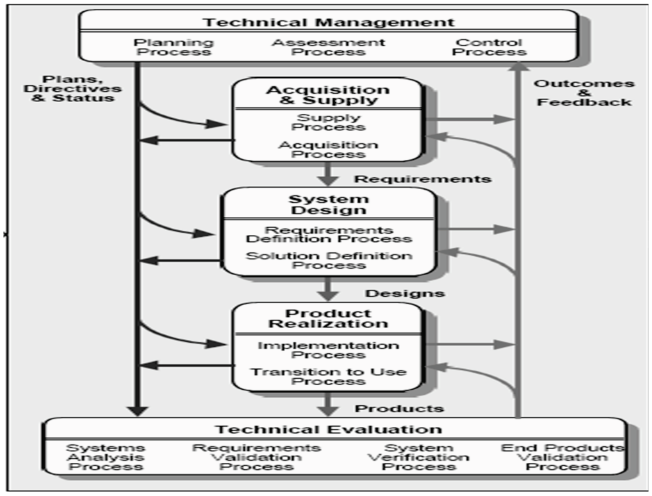


Figure 3 – The Iteration of SE Processes [25]

The International Electrotechnical Commission (IEC) and the ISO produced ISO/IEC 15288: *Systems Engineering--System Life Cycle Processes.* They also produced ISO 19760: *A Guide for the Application of ISO 15288* (IEEE, 2008). Organizations that represent the practitioners have largely accepted these efforts of standardization. IEEE has adopted the ISO standard as its own IEEE 15288. In addition, they have created IEEE 1220 to standardize the application of systems engineering processes. ISO has now accepted the IEEE 1220 as their standard (ISO/IEC, 2007). The INCOSE Handbook adheres to the technical processes codified in the ISO/IEC 15288 (INCOSE, 2007). These organizations are very active in developing and ratifying global SE standards. They recommend processes that guide systems engineering throughout the system lifecycle and among all specialties and stakeholders. ISO has just updated and released ISO 15288:2008. They are also working towards a more formal interface management that is fully part of configuration management (IEEE, 2008).

Within the DoD, there is active work to update AFI 63-1201: *Life Cycle Systems Engineering* to be consistent with ISO 15288 (Bausman, 2008). The DoD Directive (DoDD) 5000.1 mandates the application of a systems engineering approach that optimizes performance and minimizes costs (DoD, 2003a). DoDI 5000.2, paragraph 3.9.2.2 states, “The effective sustainment of weapon systems begins with the design and development of reliable and maintainable systems through the continuous application of a robust systems engineering methodology.” (DoD, 2008).

The Defense Acquisition University (DAU) provides supplementary information regarding the SE processes as they are to be applied in the DoD. They identify 16 systems engineering processes divided into eight technical processes and eight technical management processes. The technical management processes provide control and planning to the entire design effort (DAU, 2006). The technical processes for system design (known as top-down design) consist of requirements development, logical analysis, and design solution. The technical processes for product realization (known as

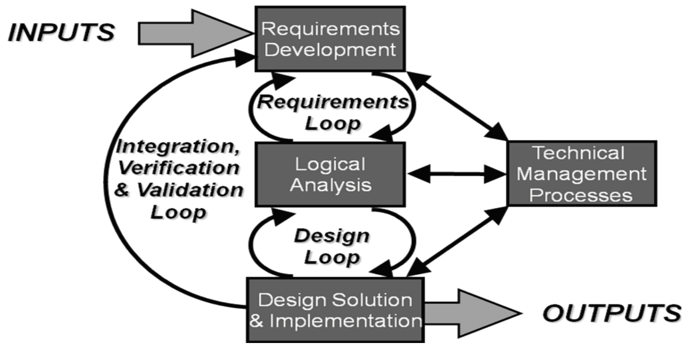


Figure 4 – DAU Concept of SE Processes Interaction
[32]

bottom-up realization) consist of implementation, integration, verification, validation, and transition (DAU, 2008). This is consistent with the EIA-632 standard (ANSI/GEIA, 2003).

A visual representation of the DAU concept of the interaction between these processes is given in Figure 4. It illustrates the iterative relationship among SE technical processes. This iterative flow is intended to be applied at all levels of the system development so that the systems engineer can define the boundary of the problem and the top-level requirements and then decompose to sufficient detail for defining feasible solutions. If the technical processes for system design are executed with tools and methods that facilitate quantitative analysis, then the technical processes for product realization can be performed quantitatively as well. Thus, our research efforts have focused on the technical processes for system design. The following paragraphs summarize the analysis performed in each process.

1.1.10 Requirements Development

The purpose of the requirements development process is to define the boundary of the problem and the top-level requirements to be satisfied. During this process, the projected mission, context, and technology readiness is evaluated. Stakeholder inputs are used to define the needs and objectives; which are refined into technical requirements. The constraints on system solutions are also identified (DAU, 2006).

1.1.11 Logical Analysis

The purpose of the logical analysis process is to improve understanding of the technical requirements and their inter-relationships. In this process, top-level requirements and constraints are decomposed and functions are allocated to system components. This creates derived technical requirements and necessary component interfaces. The process will enable the completion of system development in a logical manner (DAU, 2006).

1.1.12 Design Solution

The purpose of the design solution process is to translate the outputs of the requirements development and logical analysis process into feasible alternative solutions and to make final design decisions. This will result in a physical design of all system components capable of performing the required functions within the identified constraints. These design decisions must be objective and traceable (DAU, 2006).

1.1.13 Systems Engineering Issues

Systems engineering has generally been accepted as a necessary part of acquisition, but many studies have found evidence of problems with the execution of systems engineering processes (Bias & Mayhew, 2005; Booher, 2003; CSE, 2008a; Malone & Carson, 2003; Pew & Mavor, 2007). Sage and Rouse, in their very comprehensive *Handbook of Systems Engineering and Management*, conclude that SE processes are fundamentally sound, but the application of SE processes manifests many common problems. Their test contains a list of what they call the "most deadly systems engineering transgressions". Of the most critical, they list: a failure to develop and apply the appropriate methods to support the SE processes, a failure to design the system with consideration for the "cognitive style and behavioral constraints" that affect the users, and a failure to design the system for effective user interaction (Sage & Rouse, 1999). Designers fail to design the system for effective user interaction when they forget that users will not see the system as they do. Users will only see what is visible or represented in the user interfaces (Pew & Mavor, 2007).

Recent studies within the DoD have come to similar conclusions regarding deficiencies of systems engineering practice in military acquisition programs (CSE, 2008b; Saunders, 2005; U.S. Air Force Studies Board, 2008). The consequences of these deficiencies in execution have been costly and time consuming. A 2005 U.S. Government Accountability Office (GAO) report found that major weapon systems programs experience early cost increases by an average of 42% over original estimates and schedule slips by an average of almost 20%. Of the identified overrun causes, the GAO analysts determined that most were the result of problems that could have been discovered early in the design process (GAO, 2005). Recent DoD assessments indicate that the reason these problems are not being discovered early is that insufficient SE is applied early, requirements are not well managed, and SE tools are inadequate (CSE, 2008a).

The committees reviewing new standards for the ISO have also stated that there is a need for improved tools for requirements measurement and interface management. They desire to improve ISO standards as such tools are developed (Bausman, 2008). A recent interview with engineers at the Boeing Phantom Works reveals that defense contractors also desire better technical tools. The engineers were specifically emphatic when discussing HSI requirements. They stated that, without the ability to better capture and express these requirements quantitatively, they will never gain full consideration in program management tradeoff analysis; objective performance measures always

dominate when contracts are at stake (Graeber & Snow, 2008). Program managers continue to need actionable data earlier in the acquisition process. A U.S. Air Force HSI Office study has determined that timelines will continue to be compressed and decisions that lock in the design will increasingly need to be made before production has begun (USAF HSI Office, 2008).

The National Defense Industrial Association (NDIA) performed a study to identify areas of DoD acquisition requiring improvement. They concluded that insufficient SE is applied early in the program lifecycle; thus hindering initial requirements and architecture generation. They also concluded that the tools and methods in use are inadequate to execute the SE processes (NDIA, 2003).

In summary, there has been a plethora of studies investigating the causal factors of programmatic failures. They form a general consensus that the issue lies in the application of SE processes. Specifically, most identify a combination of the following:

- A need for sound systems engineering earlier in system development
- A need for more quantitative methods and tools to support the application of systems engineering processes
- A need for more effective management of interfaces

The Human Element of SE Issues

The challenges of integrating human components into systems are woven in the above-identified issues. This reality has led to a new DoDI 5000.2-R Sec 4.3.8 which requires comprehensive management strategies for HSI to assure effective human performance, reduce MPT requirements, and comply with all of the constraints for human operation (DoD, 2008). The DoDD 5000.1 directs the program manager to have a comprehensive plan for HSI early in the program lifecycle, a plan that will be reviewed as part of each milestone (DoD, 2003a).

This increased attention on HSI in the DoD policy reflects the understanding that demands on operators are increasing and changing in form. Studies by the Air Force HSI Office show that modern operators, with ubiquitous computer automation and augmentation, are actually experiencing more workload (USAF HSI Office, 2008). Though this seems counter-intuitive, it shows that the increase in complexity, of mission and machine, is growing faster than technological improvements can alleviate. This is being proven operationally by current unmanned aerial system (UAS) employment. The latest generation of UASs are more automated than they have ever been, but the vehicles also have more capabilities, serve in more varied roles, and execute in larger numbers than ever before (Lindlaw, 2008). This heightens the need for all domains of HSI to be considered in system development.

HSI is a critical part of system development, but it must be addressed as part of disciplined system engineering. Burns, et al. state, “System tradeoff analysis is the purview of systems engineering, and thus we posit, to be successful, HSI must live within

the systems engineering process." (Burns et al., 2005). They go on to present two axioms for HSI to address the current needs of the SE community:

- HSI must directly support the needs of acquisition
- The HSI community must provide analytic data consistent with the level of detail of design choices made within the systems engineering process.

The HSI-related system requirements are not easily quantified and the tradeoffs are often analytically complex. DoD cost studies have established the significance of such issues as 40 to 60 % of the total system's lifecycle cost (INCOSE, 2007). Air Force studies attest to the proven benefits of addressing HSI issues early in system development. Potential benefits include: reduced lifecycle cost, reduced time to fielding, shorter training cycles, improved supportability, reduced logistical footprint, and improved safety. Studies also predict that the impact of HSI issues will rapidly increase as systems and missions become more complex (USAF HSI Office, 2008). The following subsections explore what HSI methods and tools must deliver to the SE community.

HSI Empirically

Theory is essential to science. The scientific method involves creating theories that define phenomena and identify the causal relationships that explain the phenomena. In this way, theories explain what is known and predict what is unknown (Bainbridge, 2004). Work to produce unifying theories in systems engineering have been difficult. There is a common mantra that, "Systems engineering does not have an *F=ma*" (anon.); i.e., the relationship between entities are not directly connected. To date, the community primarily works in principles, heuristics, and other qualitative methods. Though a theoretical foundation of SE has proven elusive, there exist enormous volumes of empirical data regarding all domains of human systems integration. Blanchard and Fabrycky say that the improvement path for SE processes requires that they first become more quantitative and then optimized (Blanchard & Fabrycky, 2006). The engineering application of HSI data within the SE processes will achieve this.

Empirical methods are not new to human factors; anthropometric studies have been used to create ergonomic design standards for decades. The data produced in the studies allow developers to make objective decisions early in the system development process. The same is possible for other aspects of human consideration in system development.

HSI Early in System Design

The Committee on Human Factors of the National Research Council undertook a study of the current approaches for HSI in system design. They identified that the SE community needs a better set of methods and tools for early tradeoff analysis. This is a relatively novel idea. Traditionally, methods and tools of the HSI community, primarily human factors, were isolated to the design and testing of specific components (Pew & Mavor, 2007). This broadened view calls for their employment in a similar manner as

prototyping or simulation. They would be used to evaluate alternatives and minimize programmatic risk.

HSI in Accidents

The DoD has recognized the significance that HSI plays in system operation or, more specifically, in *mis*-operation. All branches of the military have agencies assigned to perform accident investigations, and these agencies are now staffed with members trained in the domains of HSI. For the U.S. Air Force, AFPD 91-2: *Safety Programs* directs the creation of all safety programs and AFI 91-202: *The US Air Force Mishap Prevention Program* directs the mishap prevention program (USAF, 1998). The USAF Safety Investigation Board (SIB) process is guided by AFI 91-204: *Safety Investigations and Reports* (USAF, 2006).

The SIB reports show that humans are still a principal component of most system accidents. For example, as shown in Figure 5, over 65% of aircraft mishaps are caused by HSI-related issues (AFSC, 2007). This persistent trend has driven safety communities toward a more focused analysis of human error and its causal relationship with mishaps. In the U.S. and allied militaries, this has led to the creation of a unified Human Factors Accident Classification System (HFACS). Systems engineers trained in the use of HFACS can translate the results of legacy system mishaps into specific and actionable data for new system development efforts.

HSI in Common System Representations

Pew and Mayor identified difficulties precluding full HSI integration in the system design process. They performed a detailed evaluation of SE, and concluded that more work needs to be done to realize an integrated, multidisciplinary set of methods and tools for human-system design. They also identified the need for a shared representation across domains and with other aspects of the system (Pew & Mavor, 2007). The act of capturing this disparate data should be integral to the design process. It would allow the

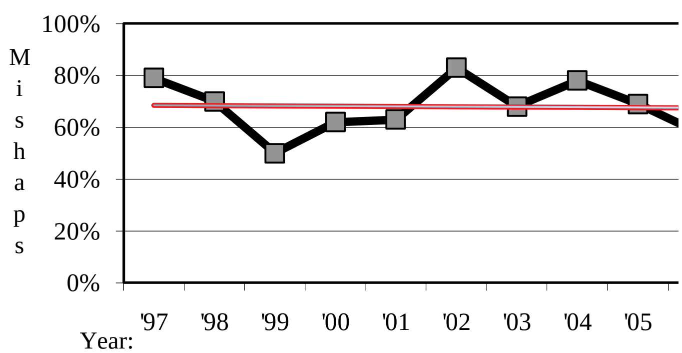


Figure 5 –Mishaps Attributed to HSI Issues [40]

ability to process greater conceptual complexity, and it would facilitate the detection of unanticipated relations and features. Without a shared representation, the design effort will continue to be stovepiped and redundant. Each domain tends to perform independent task analysis (Pew & Mavor, 2007). For the DoD, the chosen common representation has been named the DoD Architectural Framework. Hardman, et al, address the challenge of integrating HSI issues within that that recently revised framework (2008).

HSI in Interface Management

An interface is a boundary or point common to two or more entities at which necessary information flow takes place (Booher, 2003). The DoD recognizes that interface management, including the user interface, is a key to effective system integration (DAU, 2006). Maier argues that one of the key contributions that systems engineering should make to a program is an attention on the system's interfaces. He offers the following heuristic, "The greatest leverage in system architecting is at the interfaces. The greatest dangers are also at the interfaces." (Maier, 1999). This heuristic is affirmed by evidence cited in the text *Designing for the User Interface*, which illuminates the high stakes of this facet of design (Shneiderman & Plaisant, 2005). The "stakes" to the commercial sector is market share. Dray states that there is a direct tradeoff between lifecycle costs and investment in the user interface. She cites a company project in which an improved user interface on a large-scale internal application resulted in a 32% overall rate of return stemming from a 35% reduction in training and a 30% reduction in supervisory time (Dray, 1995). The stakes to the military sector are tactical advantage. Given this criticality, designers desire quantitative user interface requirements that can be objectively considered in tradeoff analysis.

Summary

As shown, HSI is central to the most challenging issues of system development. Systems engineers need to pursue better methods and tools to incorporate HSI into the overall technical approach. Key insight can be gained early in the requirements development process if legacy system mishaps are properly studied. As part of the logical analysis process, sound decision-making can be aided by more quantitative methods. Lastly, the design solution process can be improved by methods and criteria that enable the analysis of user interfaces earlier in system development.

References

AFSC, "USAF aviation safety year in review FY2006," Air Force Safety Center, Kirkland AFB, NM, 2007.

AIRPRINT, "Building Blocks of the Airmen Performance Integration Program," vol. 2007, 2005.

ANSI/GEIA, "EIA-632: Processes for engineering a system," ANSI/GEIA, 2003.

Bainbridge, W. S. Berkshire Encyclopedia of Human-Computer Interaction. Great Barrington, Mass.: Berkshire Pub. Group, 2004, pp. 958.

Bausman, K. "Systems engineering process improvement," The Center for Systems Engineering, Wright-Patterson AFB, OH, 2008.

Bias, R. G. and D. J. Mayhew, Cost-Justifying Usability: An Update for an Internet Age. Boston: Morgan Kaufman, 2005, pp. 660.

Blanchard, B. S. and W. J. Fabrycky, *Systems Engineering and Analysis.* ,4th ed.New Jersey: Pearson Prentice Hall, 2006,

Booher, H. R. *Handbook of Human Systems Integration.* , 1st ed. Hoboken, NJ: John Wiley & Sons, Inc., 2003,

Burns, J., J. Gordon, M. Wilson, M. Stretton, and D. Bowdler, "A framework for applying HSI tools in systems acquisition, paper no. 2281," in *2005 Interservice/Industry Training, Simularion, and Education Conference,* 2005,

CSE, "Systems engineering fundamentals," The Center for Systems Engineering, Wright-Patterson AFB, OH, 2008.

CSE. The current state of systems engineering. The Center for Systems Engineering, Wright-Patterson AFB, OH. 2008.

DAU, "SYS 201: Intermediate systems planning, research, development and engineering," Defense Acquisition University, Ft Belvoir, VA, 2008.

DAU. Defense Acquisition Guidebook. (2006), Available: http://akss.dau.mil/dag/welcome.asp

Dekker, S. *Ten Questions about Human Error.* Burlington, VT: CRC, 2004,

DoD, "DoD directive 5000.1: The defense acquisiton system," US Department of Defense, Washington, D.C., Tech. Rep. DoD Directive 5000.1, 2008.

DoD, "DoD instruction 5000.02: Operation of the defense acquisition system," US Department of Defense, Washington, D.C., 2008.

DoD, "MIL-STD-1472F(1): Human engineering," U.S. Department of Defense, 2003.

DoD, *MIL-HDBK-759C: Human Engineering Design Guidelines.* Washington, D.C.: US Deptment of Defense, 1998,

Dray, S., "The Importance of Designing Usable Systems," Interactions, vol. 2, pp. 17-20, 1995.

GAO, "Assessments of selected major weapon programs," U.S. Government Accountability Office, Washington, D.C., GAO-05-301, June, 2005.

Graeber, D. and M. Snow, "Interview with Boeing Phantomworks Engineers on HSI requirements," 5 May. 2008.

Hardman, N. S., J. Colombi, D. Jacques and J. Miller, "Human Systems Integration within the DoD Architecture," Proceedings of the 2008 Industrial Engineering Research Conference, 2008.

Harris, D. and H. C. Muir, Contemporary Issues in Human Factors and Aviation Safety. , vol. 50, Burlington, VT: Aldershot, 2005, pp. 963-965.

IEEE, "Systems and Software Engineering - System Life Cycle Processes," Institute for Electrical and Electronics Engineers, IEEE STD 15288-2008, 2008.

INCOSE, Systems Engineering Handbook, Ver. 3.1 International Council on Systems Engineering, 2007. International Council on Systems Engineering, 2007.

ISO/IEC, "Systems Engineering - Application and Management of the Systems Engineering Process," ISO/IEC 26702 and IEEE Std 1220-2005, 2007.

JCS, "Joint capabilities integration development system," Office of the Chairman of the Joint Chiefs of Staff, Washington, D.C., Tech. Rep. CJCSI 3170.01F, 1 May, 2007.

Lindlaw, S. "Drones' Big Enemy Is Human Mistakes," *Arizona Daily Star,* pp. 1, 26 August, 2008. 2008.

Maier, M. W. "Architecting principles for system-of-systems," John Wiley & Sons, Inc., New York, NY, Tech. Rep. CH1-460, 1999.

Malone, T. B. and F. Carson, "HSI Top Down Requirements Analysis," Nav. Eng. J., vol. 115, pp. 37-48, 2003.

Mayhew, D. J. *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design.* San Francisco, CA: Morgan Kaufmann Publishers, 1999, pp. 542.

NDIA, "Top five systems engineering issues in defense industry," National Defense Industrial Association, Systems Engineering Division, Arlington, VA, January 2003, 2003.

Norman, D. A. *The Design of Future Things.* Jackson, TN: Basic Books, 2007,

Pew, R. W. and A. S. Mayor, Human-System Integration in the System Development Process. Washington, D.C.: National Research Council, 2007.

Reason, J. *Human Error.* , 1st ed. New York: Cambridge University Press, 1990,

Sage A. and W. B. Rouse, *Handbook of Systems Engineering and Management.* New York: John Wiley & Sons, Inc., 1999.

Saunders, T. "System-of-systems Engineering for Air Force Capability Development," USAF Scientific Advisory Board, SAB-TR-05-04, 2005.

Shneiderman, B. and C. Plaisant, *Designing the User Interface.* 4th ed. Boston, MA: Addison-Wesley, 2005.

U.S. Air Force Studies Board, "Pre-milestone A and early-phase systems engineering: A retrospective review and benefits for future air force systems acquisition," The National Academies Press, Washington, D.C., 2008.

USAF HSI Office, "Briefing to the Secretary of the Air Force," U.S. Air Force HSI Office, Arlington, VA, 2008.

USAF, "AFI 91-202: Air force safety investigation requirements," Office of the Secretary of the Air Force, Washington, D.C., AFI 91-202, 1998.

USAF, "AFI 91-204: Safety investigations and reports," Office of the Secretary of the Air Force, Washington, D.C., AFI 91-204, 2006.

Appendix E5: Persons in the Processes: HSI in Early System Development

1. Introduction

Developers do not have the means to quantitatively integrate human considerations into system development. Though human systems integration (HSI) is a growing segment of systems engineering (SE) literature, studies reveal that many projects still fall short of the system effectiveness that is achievable if human components are fully integrated in the systems engineering processes (Bainbridge, 2004; Bias & Mayhew, 2005; Booher, 2003; CSE, 2008a; Dekker, 2004; Dray, 1995; GAO, 2005; Harris & Muir, 2005; Malone & Carson, 2003; Mayhew, 1999; Norman, 2007; USAF HSI Office, 2008).

Systems engineers develop systems using prescribed processes. These are interacting activities which transform inputs into outputs (ISO, 2005). Various communities have established standards to formalize SE processes. "Standards are meant to provide an organization with a set of processes that, if done by qualified persons using appropriate tools and methods, will provide a capability to do effective and efficient engineering of systems." (DAU, 2006). Standards delineate *what* needs to be done, but they generally do not dictate *how* to do it. The International Council on Systems Engineering (INCOSE) SE Handbook expands on the processes and process groups defined by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) in ISO 15288 (INCOSE, 2007). As it states in the first chapter, this standard is intended to be supported by methods and tools developed for particular organizations (ISO, 2008).

This paper proposes how to form an HSI methodology that is more multidisciplinary and empirically-based. Such an approach would address the criticisms of current methods; that they are too subjective, and that they do not take full advantage of the depth of data on human capabilities and limitations (Burns et al., 2005). Our methodology is intended to be used in the context of process activity roadmaps such as those described in the INCOSE SE Handbook, IEEE 1220, or ISO 15288 (IEEE, 2008; INCOSE, 2007; ISO/IEC, 2007). We first identify those current system development processes.

The SE technical processes defined in ISO/IEC 15288, and expanded upon in the INCOSE SE Handbook, are grouped into four process groups. One of these, the Technical Processes group, involves making technical decisions to optimize the benefits and reduce the risks during system development. These processes consist of: Stakeholder Requirements Definition, Requirements Analysis, Architectural Design, Implementation, Integration, Verification, Transition, Validation, Operation, Maintenance, and Disposal (INCOSE, 2007). In Figure 1 we express these process relationships graphically in the context of the SE Vee concept. The iterative flow that is depicted is to be applied at all levels of system development so that the systems engineer can define the boundary of the

problem and the top-level requirements and then decompose those requirements to sufficient detail for defining feasible solutions.

In Figure 1, the left side of the SE Vee consists of the technical processes for system design. If these processes are executed with methods and tools that facilitate quantitative analysis, then the technical processes for product realization, the right side of the Vee model, can be performed quantitatively as well. Thus, this paper focuses on the technical processes of the left side of the Vee.

Requirements Definition. The Stakeholder Requirements Definition process is necessary in order to define the top-level requirements. During this process, the projected mission, context, and technology readiness are evaluated. Stakeholder inputs are used to define the needs and objectives; which are refined into technical requirements. The constraints on system solutions are also identified (INCOSE, 2007).

Requirements Analysis. During the Requirements Analysis process, systems engineers improve the understanding of the technical requirements and their inter-relationships. In this process, top-level requirements and constraints are decomposed. System functions are then defined and allocated to system components. This creates derived technical requirements and necessary component interfaces. Ideally, the progression of these iterative steps is captured in an integrated common representation. This process enables the completion of system development in a logical manner (INCOSE, 2007).

Architectural Design. The Architectural Design Solution process translates the outputs of the previous processes into feasible alternative solutions used for the final design decisions. This results in a physical design of all system components capable of performing the required functions within the identified constraints. These design decisions must be objective and traceable (INCOSE, 2007).

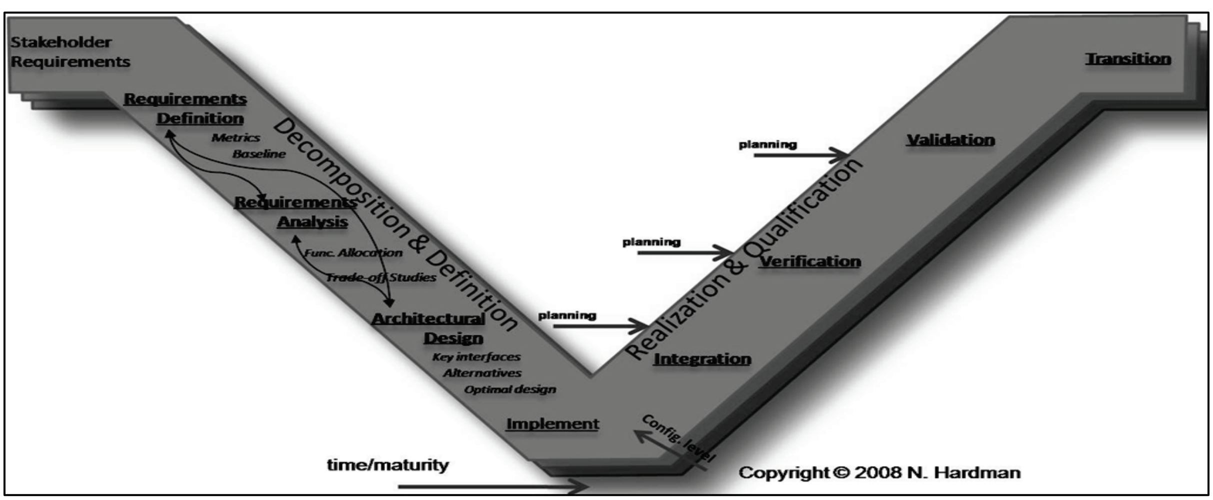


Figure 1. SE Processes on the SE Vee Model

Current Systems Engineering Issues

Systems engineering has generally been accepted as an essential part of acquisition, but many studies have found evidence of problems with the execution of the aforementioned systems engineering processes (Bias & Mayhew, 2005; Malone & Carson, 2003; Pew & Mavor, 2007). Sage and Rouse, in their comprehensive *Handbook of Systems Engineering and Management*, conclude that SE processes are fundamentally sound, but the application of SE processes manifests many common problems. They list their "most deadly systems engineering transgressions". Of the most critical, they list: a failure to develop and apply the appropriate methods to support the SE processes, a failure to design the system with consideration for the "cognitive style and behavioral constraints" that affect the users, and a failure to design the system for effective user interaction (Sage & Rouse, 1999). Designers commit these failures when they create designer-centered systems; that is, they forget that users will not see the system with the same view. Users will only see what is presented to them at the user interfaces (Pew & Mavor, 2007). The new standards reviewing committees for the ISO have also stated that there is a need for improved tools for requirements measurement and interface management. They desire to improve ISO standards as such tools are developed (Bausman, 2008).

Recent studies within the United States Department of Defense (DoD) have also found inadequate systems engineering application in military acquisition programs (Bausman, 2008; Saunders, 2005; U.S. Air Force Studies Board, 2008). A National Defense Industrial Association (NDIA) study identified areas of DoD acquisition requiring improvement. They concluded that insufficient SE is applied early in the program lifecycle, hindering initial requirements and architecture generation. They also concluded that current tools and methods are inadequate to execute the SE processes (NDIA, 2003). The consequences of these deficiencies in execution have been costly and time consuming.

A 2005 U.S. Government Accountability Office (GAO) report, found that major weapon systems programs averaged cost increases of 42% above original estimates and schedule slips of almost 20%. Of the identified overrun causes, the GAO analysts determined that most were the result of problems that could have been discovered early in the design process (GAO, 2005). Recent assessments by the Office of the US Secretary of Defense agree with the NDIA study and state that the reason these problems are not being discovered early is that insufficient SE is applied, requirements are not well-managed, and SE tools are inadequate (CSE, 2008a).

A recent interview with engineers from a major U.S. company reveals that system developers recognize the need for better technical tools. The engineers were specifically emphatic when discussing human-related requirements. They stated that, without the ability to better capture and express these requirements quantitatively, they will never gain full consideration in program management tradeoff analysis; objective performance measures always dominate when contracts are at stake (Graeber & Snow, 2008). Program managers continue to need actionable data early in the acquisition process. An

Air Force HSI Office study has determined that timelines will continue to compress and decisions that lock in design features will increasingly be made earlier in the lifecycle (USAF HSI Office, 2008).

In summary, these and other studies have investigated the causal factors of programmatic failures. They form a general consensus that the following issues exist in the application of SE technical processes:

1. sound SE *earlier* in system development
2. more SE methods and tools that provide *quantitative and actionable* data
3. more complete management of *interfaces*

HSI in SE Issues

Human components are an integral part of almost all systems, and aspects within HSI are at the heart of many of the identified systems engineering issues. The DoD recognizes this and has revised the DoD Instruction (DoDI) 5000.2-R Sec 4.3.8 to require comprehensive management strategies for HSI to assure human performance, reduce manpower, personnel, and training requirements and comply with all of the constraints for human operation (DoD, 2003a). The new DoD Directive (DoDD) 5000.1 directs the program manager to have a comprehensive HSI plan for early in the program lifecycle; a plan that is reviewed at each milestone. This increased attention on HSI in the DoD policy reflects the understanding that demands on operators are increasing and changing in form. Studies by the Air Force HSI Office show that modern systems operators, surrounded by ubiquitous computer automation and augmentation, are actually experiencing greater cognitive workload (USAF HSI Office, 2008). Though this seems counter-intuitive, it shows that the increase in complexity, both of mission and machine, is growing faster than technological improvements can alleviate. This is being proven out in operation in Unmanned Aerial System (UAS) deployment. The latest generation of UASs are the most automated ever, but they are also have more capabilities, serve in multiple roles, and fly in larger numbers than ever before (Lindlaw, 2008). This heightens the need to properly embed all domains of HSI in the system development processes.

HSI Domains

Though there is no universally accepted delineation of HSI domains, there is much in common among the definitions found in current literature. The key components have been extracted and written in terms that enable a system engineer to clearly categorize system requirements by domain and to perform tradeoff analysis between domains. These definitions draw heavily upon those put forth by INCOSE and the Human Effectiveness Directorate of the Air Force Research Laboratory (AIRPRINT, 2005; DoD, 2008; INCOSE, 2007).

Manpower. The Manpower domain determines the number and type of personnel required to operate and support a system. Support includes functions such as

maintenance, sustainment, and training. Many civilian organizations call this "human resources".

DoD direction on manpower estimates for major defense acquisition programs is extensive. Program managers must coordinate with the manpower community, and the final manpower estimate is reviewed by the Under Secretary of Defense for Personnel and Readiness (DoD, 1999).

Personnel. The Personnel domain determines the knowledge, skills, and abilities (KSAs) and the physical, cognitive and sensory capabilities required of the humans in the system. The personnel community defines these parameters for the system and determines how to best obtain and maintain an adequate pool of qualified people. Some military organizations call it "personal capabilities", and in civilian organizations this domain is inter-related with human resources.

Training. The Training domain determines the necessary infrastructure and system components to provide system personnel with the requisite attributes (KSAs) for optimal system performance. This includes individual and unit training programs, training systems, and retraining schedules.

Human Factors. The Human Factors (HF) domain addresses how to incorporate human characteristics and limitations into system design for optimal usability. A primary concern for HF is the creation of effective user interfaces. Issues in this domain are often divided into the following categories:

Cognitive— e.g., response times, level of autonomy, cognitive workload limitations

Physical— e.g., ergonomic control design, anthropometric accommodation, workload limitations

Sensory— e.g., perceptual capabilities, such as sight, hearing, or tactile

Team dynamic—e.g., communication and delegation, task sharing, crew resource management

Much of U.S. industry calls this "human factors engineering (HFE)" and European and Asian organizations generically refer to it as "ergonomics". The methods and tools of this domain are the most mature of all the HSI domains.

System Safety. The System Safety domain evaluates the characteristics and procedures of systems in order to minimize the potential for accidents. Safety studies affect system design by advocating features that eliminate hazards when possible and manage those unavoidable hazards. Such features include sub-systems for: system status, alert, backup, error recovery, and environmental risk.

Survivability. The Survivability domain evaluates the characteristics and procedures of systems that can reduce the probability of attack or fratricide, as well as minimizing system damage and injury if attacked.

Health. The Health domain evaluates the characteristics and procedures of systems that create significant risks of injury or illness to humans. Sources of health hazards include: noise, temperature, humidity, CBRNE (chemical, biological, radiological, nuclear, and explosive) substances, physical trauma, and electric shock.

Habitability. The Habitability domain evaluates the characteristics and procedures of systems that have a direct impact on personnel effectiveness by maintaining morale, comfort, and quality of life. These characteristics uniquely include: climate control, space layout, and support services.

Environment. The Environment domain evaluates the system in the medium for operation. Consideration is made to protect the environment from system manufacturing, operations, sustainment, and disposal activities. In some communities this domain is not considered part of HSI, but rather of systems engineering as a whole.

The Challenges of HSI

HSI-related system requirements are not easily quantified and the tradeoffs are often analytically complex. What is clear is the significance of such issues. DoD cost studies estimate that 40 to 60 % of the total system's lifecycle cost is determined by decisions in the HSI domains (INCOSE, 2007). Air Force studies attest to the proven benefits of addressing HSI issues early in system development. Potential benefits include: reduced lifecycle cost, reduced time to fielding, shorter training cycles, improved supportability, reduced logistical footprint, and improved safety. Studies also predict that the impact of HSI issues will increase as systems and missions become more complex (USAF HSI Office, 2008). HSI must be addressed as part of disciplined system engineering. Burns, et al. state, “System tradeoff analysis is the purview of systems engineering, and thus we posit, to be successful, HSI must live within the systems engineering process.” (Burns et al., 2005). They go on to present two axioms for what HSI must be in order to address the needs of the SE community:

- HSI must directly support the needs of acquisition.
- The HSI community must provide analytic data consistent with the level of detail of design choices made within the systems engineering process.

The National Research Council recently reviewed the current approaches for HSI in system design. They recommended that researchers develop better ways to incorporate human-related issues into early system tradeoff analysis. Traditionally, methods and tools of the HSI community were primarily applied to the design and testing of specific components (Pew & Mavor, 2007). This broadened view calls for an approach similar to prototyping or simulation. This calls for new methods that can be used to evaluate system alternatives and minimize programmatic risk. Systems engineers need better methods and tools to incorporate HSI into the overall technical approach. These methods and tools must help them better perform requirements elicitation, function allocation, tradeoff analysis, and design optimization. We next propose a methodology for use early and throughout the SE Technical Processes.

A Better Methodology for HSI

The SE processes can be more effective if the before mentioned issues are satisfactorily addressed. Published SE standards are essential for specifying the design processes, but they need methods to provide the means for successful system development. Methods are systematic ways of doing things, and some methods involve tools to perform specific steps of the process. A complete methodology requires more than a set of these methods and tools; it requires a unifying basis that underlies the whole approach. New system development will encounter many key decisions that will commit the design effort in a nearly irreversible course. We propose some needed elements of an improved methodology; one that will enable sound decision making early in the technical processes.

Approach HSI Empirically

Theory is essential to science. The scientific method involves creating theories that define phenomena and identify the causal relationships that explain the phenomena. In this way, theories explain what is known and predict what is unknown (Bainbridge, 2004). Work to produce unifying theories in systems engineering have been difficult. There is a common mantra that, "Systems engineering does not have an F=ma" (anon.); the relationship between entities are not directly connected. To date, the community primarily works with principles, heuristics, and other qualitative methods. Though a theoretical foundation of SE has proven elusive, there exist enormous volumes of empirical data regarding all domains of human systems integration. Blanchard and Fabrycky say that the improvement path for SE processes requires that they first become more quantitative and then optimized (Blanchard & Fabrycky, 2006). Though this approach may not be feasible for all issues, there is still great potential in the proper engineering application of empirical data within the SE processes.

Empirical methods are not new to human factors; anthropometric studies have been used to create ergonomic design standards for decades. The data produced in the studies allow developers to make objective decisions early in the system development process. The same is possible for other aspects of human consideration in system development; albeit with much less of a direct connection. For example, in (Hardman, Colombi, Jacques, & Hill et al., 2008b) we investigated the process for menu layouts in multi-function displays. Current design efforts rely on designer intuition followed by extensive user testing. We discovered that the effectiveness of a given menu layout could be accurately predicted using existent human subjects data. This gives designers a quantitative tool for early design decisions, and it greatly reduces the necessary iterations of costly user testing. The following sub-sections give two additional propositions of how to improve SE with a more empirical approach.

Quantify Predictions. Modern perspectives in accident investigation follow the theories of Dr. J. Reason who proposed that a hazard becomes an accident through the ill-fated alignment of certain latent conditions in the systems that are in place to prevent such accidents (Reason, 1990). This concept is represented in Figure 2. The data

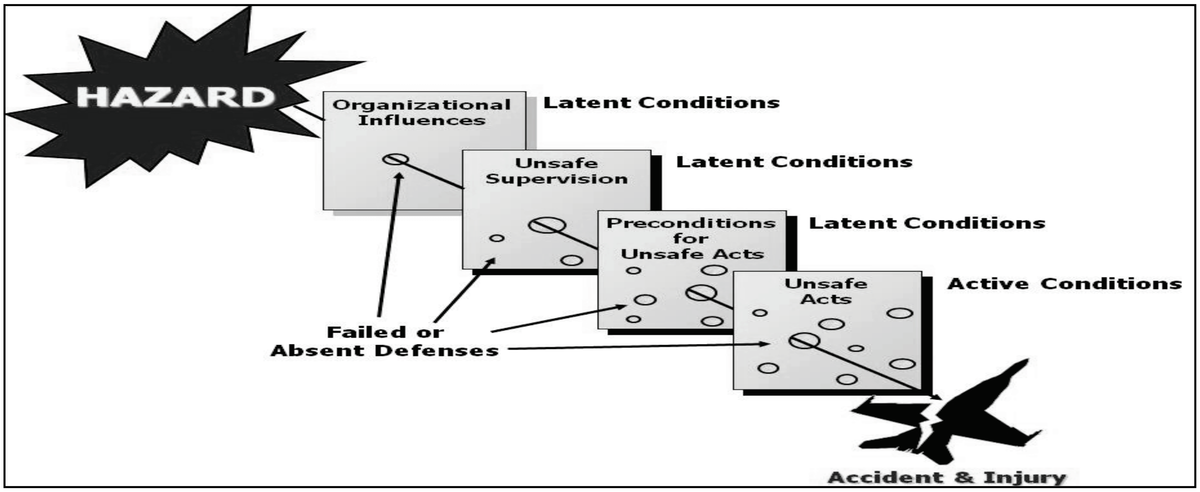


Figure 2. Accident Causation Theory
(Src: AFSC, 2007)

gathered by accident investigation boards on legacy systems can reveal valuable insight for the design of the next generation of systems, but only if design teams know how to use that data. The DoD has recognized the significance that HSI plays in military operations, or more specifically, in *mis*-operations. The U.S. military, and most NATO forces, have independent safety organizations responsible for accident investigation and reporting. For the U.S. Air Force, AFPD 91-2: *Safety Programs* directs the creation of all safety programs and AFI 91-202: *The US Air Force Mishap Prevention Program* directs the mishap prevention program. The USAF Safety Investigation Board (SIB) process is guided by AFI 91-204: *Safety Investigations and Reports* (USAF, 2006). These regulations give specific instruction for investigating how issues in the domains of HSI relate to specific events.

The archives of legacy system mishaps provide a large database of known problems. Studies of human error in previous mishaps can reveal how and why the human-machine interaction may fail during similar operational activities. Even very novel systems have similarities, either in structure or use, to legacy systems. Knowing what design issues have plagued past operators can guide the generation of requirements to address the identified issues. This will contribute to measures of: total system reliability, the impact of automation, and human-in-system parametrics. In a general sense, these provide quantitative justification for giving HSI domains proper attention. For example, as shown in Figure 3, aircraft mishap investigations reveal that over 65% of aircraft mishaps still involve HSI-related causal factors (AFSC, 2007). With more specific analysis, design decisions in early system development can be made using quantitative data.

Quantify User Interface Requirements. An interface is a boundary or point common to two or more entities at which necessary information flow takes place (Booher, 2003). The DoD recognizes that interface management, including the UI, is a key to effective system integration (DAU, 2006). Maier argues that one of the key contributions that SE should make to a program is an attention to the system's interfaces. He offers the

following heuristic, "The greatest leverage in system architecting is at the interfaces. The greatest dangers are also at the interfaces." (Maier, 1999). Maier's heuristic aligns with evidence cited in the text, *Designing for the User Interface* regarding the interfaces of computer and machine (Shneiderman & Plaisant, 2005). Dray identifies a direct connection between lifecycle costs and investment in user interfaces. She cites a company project in which an improved user interface on a large-scale internal application resulted in a 32% overall rate of return stemming from a 35% reduction in training and a 30% reduction in supervisory time (Dray, 1995). Savings such as these continue throughout the life of a system and become a significant factor in controlling total ownership cost. Given this criticality, systems engineers must have a methodology that properly focuses on UI design.

As computers become more ubiquitous, systems engineers recognize the immense significance of the interaction of operators and computers. Formal study of this has matured and expanded in perspective over the last three decades. It is now generally referred to as human-computer interaction (HCI). Our composite definition is that the field of HCI is "a field of study that seeks to improve the relations between users and computers by making computers more usable, intuitive, and accommodating of human capabilities and limitations." Many publications use HCI-related terms in an inconsistent manner. The DoD has listed standard terms and a taxonomy for these terms in the military handbook (MIL-HDBK) -1908B: *DoD Handbook on Definitions of Human Factors Terms* (DoD, 1999) and MIL-STD-1472: *Human Engineering* (DoD, 2003b). Several professional societies have also defined their standard terminology, but none has gained universal acceptance.

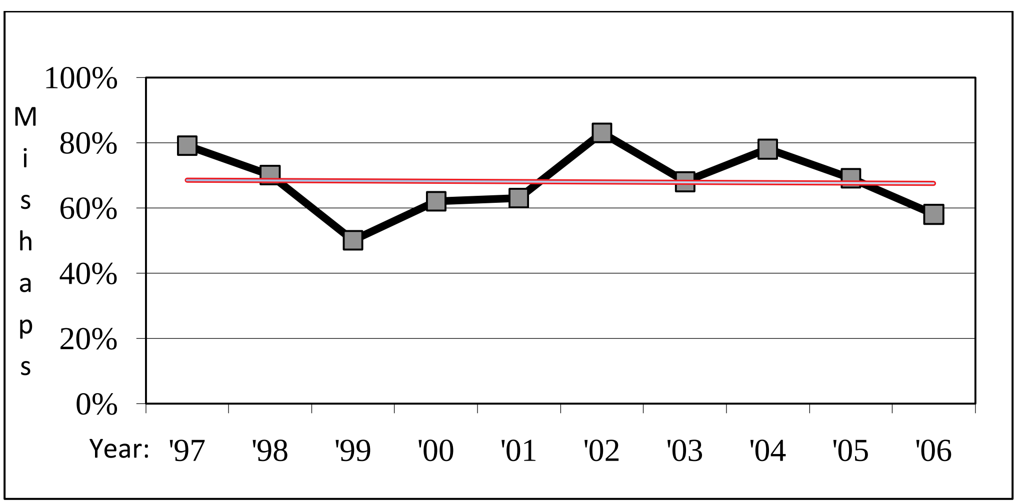


Figure 3. Annual Percentage of Mishaps Attributed to HSI Issues
(Src: AFSC, 2007)

Though the terminology varies by community, the concepts are similar. The central emphasis of HCI, by that or any other name, is the design of effective user interfaces (UIs); that is, the multi-modal exchanges between a human being and hardware. These interfaces facilitate interaction between human cognition and software logic. This is a critical part of HSI. Figure 8 4 is a conceptual depiction of human factors, HCI, and UI design. The theories and methods of the HCI community can be very useful in environments where the demand for highly effective operator performance is paramount.

HCI requirements are less concrete than other system requirements. HCI experts call the measure of a user interface its *usability*. Various sources expand on the concept of usability in different ways, but it is common to define the measure using the following five components (Wickens et al., 2004):

<u>Reliability --</u> The frequency of errors, the prevention of catastrophic errors, and the ability to recover from errors.
<u>Efficiency</u> -- The level of productivity that can be achieved once learning has occurred.
<u>Learnability</u> -- The amount of training necessary before the user can be productive.
<u>Memorability</u> -- The ability for an infrequent user to maintain proficiency.
<u>Satisfaction</u>: -- The subjective experience of the user.

These last three components are unique to the user interface. Because we cannot yet simply program humans, interface learning time and recall ability are essential interface metrics. Standard HCI texts, such as the *Berkshire Encyclopedia of HCI*, address how to measure the five components of usability (Bainbridge, 2004). With the use of the proper benchmarks and surveys, these usability metrics can be standardized and quantified for use in system development. This is discussed in (Hardman, Colombi, Jacques, & Hill et al., 2008b). While usability evaluations make UI analysis quantifiable, designers have a need to predict the potential usability of a configuration even earlier in system

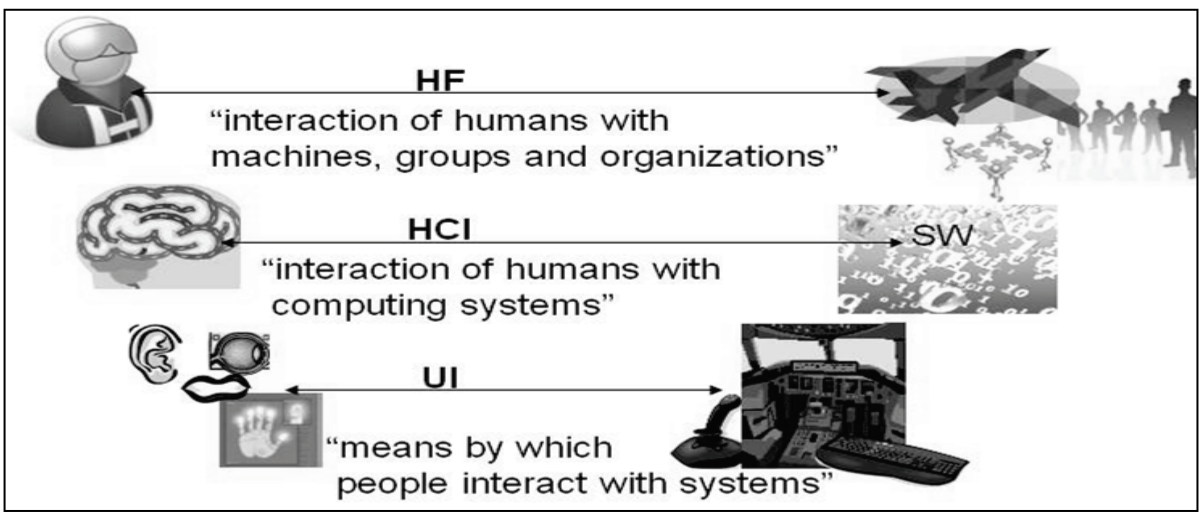


Figure 4. Human Factors Domain Components

development. Usability analysis cannot be performed until actual systems or prototypes are developed, but empirical human data models can make projections of UI usability much earlier. This is akin to the use of wind tunnels to predict the performance of an aircraft wing even before an actual one is built. Such information could equip systems engineers to effectively address HCI early in system development.

Develop a Requirements and Constraints Paradigm

When incorporating HSI domains into SE technical processes, not all domains should be addressed the same. The human factors, manpower, personnel, and training domains can be effectively managed as design variables that interact with other system requirements. Issues within these domains are often inextricably connected, and system tradeoff analyses must include those inter-relations. For example, as described above, program managers must make early estimates regarding manpower requirements. However, the manpower estimate for a system under development is highly sensitive to decisions made in the personnel, training and human factors domains. This means the manpower estimate must concur with the analysis contained in many tangential documents throughout the system development process.

Conversely, the domains of safety, survivability, health, habitability, and environment form constraints on the system. The analysis of requirements and constraints due to human considerations is a multi-dimensional optimization problem. A useful perspective is to consider the analog of a physical volume as portrayed in Figure 95. Within the context of the system's intended mission, the constraints limit the size and shape of the trade space. The necessary performance of the system creates an analogous volume with manpower, personnel, and training dimensions. Systems engineers must meet the mission requirements, the volume, in such a way that it fits within the contextual constraints, the space. Human factors issues have the effect of either increasing the volume or filling it. Poor human consideration can increase demands in the manpower, personnel, and training dimensions; alternatively, improved human factors can be effective force multipliers by reducing operator workload. This paradigm defines an approach that can make tradeoff analysis more complete, context-aware, and objective.

Integrate with SE Technical Process

The purpose for an improved HSI methodology is to provide actionable data to make timely design decisions. To do that, it must be embedded within the SE technical processes for design.

Requirements Definition. As mentioned, HCI practitioners study how to best capture system interactions. The HCI community can use these methods to predict areas for design emphasis in order to avoid what Hoffman and Elm call the "pitfalls of designer-centered designs" (Hoffman & Elm, 2006). Even more lucid insights are possible if prototypical problems have been cataloged. The Requirements Definition process is the proper time to perform an empirical study of legacy system mishaps involving human

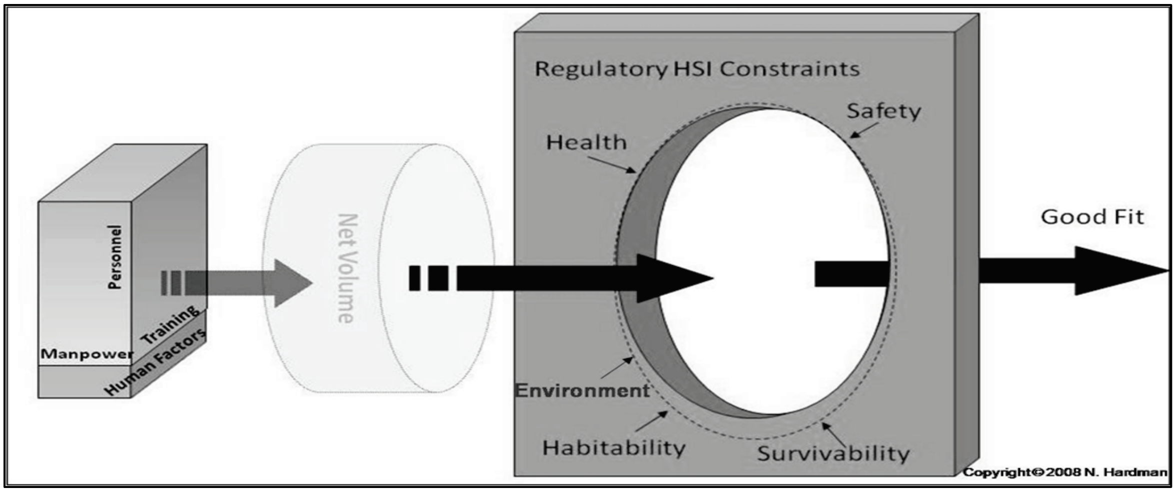


Figure 5. HSI Requirements and Constraints

error as a causal factor. This is similar to the failure analysis already used to provide feedback to the acquisition community regarding structural and propulsion aircraft components. This may also identify additional human subjects research that may be needed to prepare for the following processes.

Requirements Analysis. During the Requirements Analysis process, a significant amount of effort is required for decomposition and function allocation. A function is a logical unit of behavior of a system (Blanchard & Fabrycky, 2006). Function allocation is "a procedure for assigning each system function, action, and decision to hardware, software, operators, maintainers, or some combination of them" (NASA, 2007). This is normally a process that involves more art than science. Humans, as components of the system, must be a part of function allocation decisions.

When it concerns humans and computers, function allocation involves the study of automation. Many function allocation decisions are made at the micro level when components are selected. This often leaves the designer with a limited number of highly constrained explicit allocation decisions. Function allocation influences all domains of HSI because they affect the number and location of operators. Therefore, function allocation decisions precede other SE tasks such as: task analysis, operator workload analysis, and UI design. Function allocation must be done in a manner that optimizes performance and safety. If one uses empirical data for function allocation decisions, those decisions will be more traceable and will enable easier adaptation of new missions or technology. Previous efforts in this area were highly influenced by the famous 1951 Fitt's List that itemized what computers did better versus what humans did better (Sheridan, 2000). The problem with this approach is twofold: First, a division along such lines will quickly be made obsolete by the changing landscape of technology. More significantly, this approach takes an incomplete view of the problem. At its core, the challenge is to understand the required tasks and how to support the humans that must do them. This requires more information regarding the given context and the desired

outcome. We can draw from empirical data from basic research in physiology and psychology to clarify these decisions. If done in a context-aware approach, i.e., one that accounts for mission and medium factors, this method will yield quantitative answers for automation decisions. For example, designers of unmanned aerial systems (UASs) desire to examine the automation of detect, see and avoid (DSA) tasks. They desire quantitative measures of the performance differences between humans and automated systems. In (Hardman 2005) this was analyzed using existing human subjects and sensor data.

Architectural Design. The Architectural Design process includes determining the feasibility of alternatives and ultimately choosing the final design solution. This can be a complicated sequence that must be done early and throughout system development. Usability analysis and model-based predictions can support alternative evaluation.

The proposed approach should be applied to the design of input devices. Many systems require dynamic inputs from humans. This includes manual tracking tasks such as slewing a target for object selection, panning and zooming, and the teleoperation of robotic vehicles. Systems engineers must assure that the interface between the human operators and the machine is not a performance limiting factor. These interactions have the following characteristics: the operator knows the goal, the system possesses the means to achieve the goal, there is a speed and accuracy tradeoff of communicating the goal to the system, and the operator receives feedback regarding performance. The interface must account for the parameters of both the machine and the human. Control systems texts offer very complete procedures for determining the response parameters of machines, but human-inclusive determinations are less defined.

Another area that needs to be addressed is user displays layout design. Display design begins with specifications listing all necessary information and configuration requirements. The designer must find the best layout to satisfy these specifications. The layout should follow a context-aware design paradigm in that the informatic relationships are based on what is needed for a given operational activity thread (Dix, Finlay, Abowd, & Beale, 2004). For example, for aviation applications, experimental psychologists have correlated necessary information by phase of flight (Schvaneveldt et al., 2001). This information can not only improve early design suitability evaluation, it can greatly reduce the effort required for alternative analysis. This was the impetuous for the approach, as detailed earlier, to predict display layout effectiveness (Hardman, Colombi, Jacques, & Hill, 2008a).

Summary

We have summarized the current SE challenges and examined how HSI is a pervasive factor in these challenges. We then proposed how to improve system engineering through a methodology that would address the identified challenges. Throughout the discussion, we referenced specific work being done using such a methodology. This approach can bridge empirical research on human capabilities and the challenges of SE in early system development.

References

AFSC. 2007. U.S. Air Force aviation safety year in review FY2006. Kirkland AFB, NM: Air Force Safety Center.

AIRPRINT. Building blocks of the airmen performance integration program. in 311th Human Systems Wing [database online]. Brooks City-Base, TX, 2005 [cited 10 Aug 07]. Available from http://www.brooks.af.mil/.

Bainbridge, William Sims, ed. 2004. Berkshire encyclopedia of human-computer interaction. Great Barrington, Mass.: Berkshire Pub. Group.

Bausman, Karen. 2008. Systems engineering process improvement. Wright-Patterson AFB, OH: The Center for Systems Engineering, .

Bias, Randolph G., and Deborah J. Mayhew, eds. 2005. Cost-justifying usability: An update for an internet age. Boston: Morgan Kaufman.

Blanchard, Benjamin S., and Wolter J. Fabrycky. 2006. Systems engineering and analysis. 4th ed. New Jersey: Pearson Prentice Hall.

Booher, Harold R., ed. 2003. Handbook of human systems integration. Systems Engineering and Management., ed. Andrew Sage. 1st ed. Hoboken, NJ: John Wiley & Sons, Inc.

Burns, John, Jerry Gordon, Matt Wilson, Milt Stretton, and Dan Bowdler. 2005. A framework for applying HSI tools in systems acquisition, paper no. 2281. Interservice/Industry Training, Simularion, and Education Conference, Orlando, FL.

CSE, The Center for Systems Engineering. 2008. The current state of systems engineering. Acquistion Professional Training Course. Wright-Patterson AFB, OH:, .

DAU, Defense acquisition guidebook. Ft Belvior, VA. 2006 [cited 20 Jan 09]. Available from http://akss.dau.mil/dag/welcome.asp.

Dekker, Sidney. 2004. Ten questions about human error. Burlington, VT: CRC Press.

Dix, A., J. Finlay, G. Abowd, and R. Beale. 2004. Human-computer interaction. 3rd ed. Essex: Pearson Education Limited.

DoD. 2008a. DoD directive 5000.1: The defense acquisiton system. Washington, D.C.: US Dept. of Defense.

———. 2008b. DoD instruction 5000.2: Operation of the defense acquisition system. Washington, D.C.: US Dept. of Defense, .

———. 2003. MIL-STD-1472F(1): Human engineering. 5 Dec 2003: US Dept. of Defense, MIL-STD-1472F(1).

———. 1999. MIL-HDBK-1908B: Definitions of human factors terms. Washington, D.C.: US Dept. of Defense, MIL-HDBK-1908B.

———. 1998. MIL-HDBK-759C: Human engineering design guidelines. Washington, DC: US Dept. of Defense.

Dray, Susan. 1995. The importance of designing usable systems. Interactions.

GAO. 2005. GAO-05-301: Assessments of selected major weapon programs. Washington, D.C.: US Government Accountability Office, GAO-05-301.

Graeber, D., and Michael Snow. 2008. Interview with boeing phantomworks engineers on HSI requirements. 5 may, Seattle, WA.

Hardman, Nicholas S., J. Colombi, D. Jacques, and R. Hill. 2008. Improved user interface design through graph-theoretic modeling. Proceedings of IEEE International Conference on Distributed Human-Machine Systems.

———. 2008. What systems engineers need to know about human-computer interaction. INCOSE Insight 11, (2): 19-23.

Hardman, Nicholas S. 2005. Approaches for Autonomous Vehicles in Civil Airspace, Master's Thesis, University of Tennessee.

Harris, Don, and Helen C. Muir, eds. 2005. Contemporary issues in human factors and aviation safety. Vol. 50. Burlington, VT: Aldershot.

Hoffman, R., and W. Elm. 2006. HCC implications for the procurement process. Intelligent Systems 21.

IEEE. 2008. Systems and software engineering - system life cycle processes. Institute for Electrical and Electronics Engineers, IEEE STD 15288-2008.

INCOSE. 2007. Systems engineering handbook, ed. Cecilia Haskins. Ver. 3.1 ed.International Council on Systems Engineering.

ISO 15288. 2008. A system and software engineering standard. International Organization for Standardization, ISO 15288:2008.

ISO 9000. 2005. Quality management systems--fundamentals and vocabulary. International Organization for Standardization, 9000:2005.

ISO/IEC 26702. 2007. Systems engineering - application and management of the systems engineering process (also, IEEE std 1220-2005). ISO and IEEE.

Lindlaw, Scott. 2008. Drones' big enemy is human mistakes. Arizona Daily Star, 26 August, 2008, 2008.

Maier, Mark W. 1999. Architecting principles for system-of-systems. New York: John Wiley & Sons, Inc.

Malone, T. B., and F. Carson. 2003. HSI top down requirements analysis. Naval Engineers Journal. 115.

Mayhew, Deborah J. 1999. The usability engineering lifecycle: A practitioner's handbook for user interface design. The morgan kaufmann series in interactive technologies. San Francisco, CA: Morgan Kaufmann Publishers.

Narkevicius, Jennifer M., John Winters, and Nicholas S. Hardman. 2008. Talking the talk: Cross-discipline terminology challenges. INCOSE Insight 11, (2): 25-7.

NASA. 2007. NASA/SP-2007-6105: Systems engineering handbook. REv. 1 ed. Washington, D.C.: National Aeronautics and Space Administration.

NDIA. 2003. Top five systems engineering issues in defense industry. Arlington, VA: National Defense Industrial Association, Systems Engineering Division, .

Norman, Don A. 2007. The design of future things. Jackson, TN: Basic Books.

Pew, Richard W., and Anne S. Mavor, eds. 2007. Human-system integration in the system development process. Washington, D.C.: National Research Council.

Reason, James. 1990. Human error. 1st ed. New York: Cambridge University Press.

Sage, Andrew, and William B. Rouse, eds. 1999. Handbook of systems engineering and management. New York: John Wiley & Sons, Inc.

Saunders, Thomas. 2005. System-of-systems engineering for air force capability development. USAF Scientific Advisory Board, SAB-TR-05-04.

Schvaneveldt, Roger W., Dennis B. Beringer, and John A. Lamonica. 2001. Priority and organization of information accessed by pilots in various phases of flight. 11, (3) (International Journal of Aviation Psychology): 253-80.

Sheridan, T. B. 2000. Function allocation: Algorithm, alchemy or apostasy? International Journal of Human-Computer Studies, 52, (2) (2): 203-16.

Shneiderman, Ben, and C. Plaisant. 2005. Designing the user interface. 4th ed. Boston, MA: Addison-Wesley.

U.S. Air Force Studies Board. 2008. Pre-milestone A and early-phase systems engineering: A retrospective review and benefits for future air force systems acquisition. Washington, D.C.: The National Academies Press, .

USAF. 2006. AFI 91-204: Safety investigations and reports. Washington, D.C.: Office of the Sec. USAF.

USAF HSI Office. 2008. Briefing to the secretary of the air force. Arlington, VA: U.S. Air Force HSI Office, .

Wickens, Christopher D., John Lee, Yili Liu, and Sallie Gordon-Becker. 2004. An introduction to human factors engineering. 2nd ed. Upper Saddle River, NJ: Prentice Hall.

Author Vita

Major Nicholas Hardman, USAF is a doctoral student at the Air Force Institute of Technology, Wright-Patterson AFB, Ohio. His dissertation research is focused on how to better integrate humans into system design.

Nick was born in Pendleton, Oregon. He graduated from Seattle Pacific University in 1996 with a bachelor's degree in electrical engineering and earned a research fellowship in the Boeing Control Systems Laboratory at the University of Washington. In 1997 he was commissioned into the U.S. Air Force. He completed and published his thesis work in the control of elastically-coupled systems at the University of Washington and earned a master's degree in electrical engineering.

Nick attended the US Air Force Test Pilot School and has performed flight evaluations for the first GPS-based weapons. His paper on this work which was recognized by the Air Force Flight Test Center and published by NATO's science advisory council.

While serving as flight commander of the Tactical Communications and Surveillance Test Flight, Nick oversaw multiple first flight programs in avionics and defensive systems. As an exchange officer to the Spanish Air Force, Nick oversaw the development of new UAVs for coalition forces.

In 2005, Nick completed research on the study of unmanned aerial vehicles in unrestricted airspace and earned a master's degree in aviation systems management from the University of Tennessee. In 2006 he was accepted for the doctoral program at AFIT. Upon graduation, he will assume the position of Chief Systems Engineer for the C-5 Modernization Program.

Bibliography

ACM. (2006). *ACM Overview.* Retrieved May 30, 2007, from http://www.acm.org/.

Adams, C. (2001, Feb). UAVionics in the US. *Aviation Toda* , p. 20.

AF Tech. (2009). *RQ-4A/B global hawk high-altitude, long-endurance, unmanned reconnaissance aircraft, USA.* Retrieved Aug 22, 2009, from http://www.airforce-technology.com/projects/global/.

AFMC. (2007). *AFMCI 63-1201: Implementing OSS&E and life cycle systems engineering* (Air Force Materiel Command Instruction No. 63-1201). Washington, D.C.: US Air Force.

AFSC. (2007). *USAF aviation safety year in review FY2006* (Briefing. Kirkland AFB, NM: Air Force Safety Center.

Ahuha, R., Magnanti, T., & Orlin, J. (1993). *Network flows: Theory, algorithms, and applications*. New Jersey: Prentice Hall.

AIRPRINT. (2005). *Building blocks of the airmen performance integration program.* Retrieved August 10, 2007, from http://www.brooks.af.mil/.

Albert, R., Jeong, H., & Barabasi, A. (1999). Diameter of the world wide web. *Nature*, p. 130.

Albery, W. B., & Khomenko, M. N. (2003). Differences in pilot automation philosophies in the US and russian air forces ground collision avoidance systems. *Technical Reports*, p. 9.

Andre, L., & Kukura, M. (2009). *See and avoid.* Retrieved 10 Sep 09, 2009, from http://www.smartcockpit.com/flightops.

Andrews, J. W. (1991). Air-to-air visual acquisition handbook.

ANSI/GEIA. (2003). *EIA-632: Processes for engineering a system*ANSI/GEIA.

Bainbridge, W. S. (Ed.). (2004). *Berkshire encyclopedia of human-computer interaction*. Great Barrington, MA: Berkshire Pub. Group.

Baker, A., & Grynbaum, M. M. (2009, Aug 10). Airspace above hudson a highway with few signs. *The New York Times*, p. 1.

BASI. (1991). *Limitations of the see-and-avoid principle*. Sydney: Civil Aviation Authority.

Bausman, K. (2008). *Systems engineering process improvement*. Wright-Patterson AFB, OH: The Center for Systems Engineering.

Bias, R. G., & Mayhew, D. J. (Eds.). (2005). *Cost-justifying usability: An update for an internet age*. Boston: Morgan Kaufman.

Bisantz, A., & Burns, C. (2009). *Applications of cognitive work analysis*. Boston: CRC Press.

Blanchard, B. S., & Fabrycky, W. J. (2006). *Systems engineering and analysis* (4th ed.). New Jersey: Pearson Prentice Hall.

Boff, K., Kaufman, L., & Thomas, J. (Eds.). (1986). *Handbook of perception and human performance*. New York: Wiley & Sons.

Booher, H. R. (Ed.). (2003). *Handbook of human systems integration* (1st ed.). Hoboken, NJ: John Wiley & Sons, Inc.

Bourbaki, N. (2003). *Elements of mathematics: Functions of a real variable*. New York: Springer.

Bowman, J. (2009). In Hardman N. S. (Ed.), *Interview with personnel of AFRL future unmanned systems*. Wright-Patterson AFB, OH.

Buede, D. M. (2000). *The engineering design of systems* (1st ed.). New York: John Wiley & Sons, Inc.

Burns, J., Gordon, J., Wilson, M., Stretton, M., & Bowdler, D. (2005). A framework for applying HSI tools in systems acquisition, paper no. 2281. *2005 Interservice/ Industry Training, Simulation, and Education Conference,* Orlando, FL.

Calhoun, G., Herron, E., Reising, J., & Bateman, R. (1980). *Evaluation of factors unique to multi-function Controls/Displays devices* No. Tech Rep #80-3131). Wright-Patterson AFB, OH. AFFDL.

Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., p. 47.

Celik, M., & Cebi, S. (2009). Analytical HFACS for investigating human errors in shipping accidents. *Journal of Accident Analysis and Prevention, vol. 41*, pp. 66-75.

Chamlou, R., Love, W. D., & Moody, C. (2008). Exploration of new algorithms for airborne collision detection and avoidance to meet NextGen capabilities. *Digital Avionics Systems Conference. DASC '08. IEEE/AIAA*. St Paul, MN. pp. 2.D.5-1 to 2.D.5-13.

Crandal, B., & Klein, G. A. (2006). *Working minds: A practitioner's guide to cognitive task analysis*. Boston: The MIT Press.

CSE. (2008a). *The current state of systems engineering*. Wright-Patterson AFB, OH: The Center for Systems Engineering.

CSE. (2008b). *Systems engineering fundamentals* (Educational Briefing. Wright-Patterson AFB, OH: The Center for Systems Engineering.

DAU. (2006). *Defense acquisition guidebook*. Retrieved Sep 1, 2009, from http://akss.dau.mil/dag/welcome.asp.

DAU. (2008). *SYS 201: Intermediate systems planning, research, development and engineering* (Acquisition Professional Training Course Materials). Ft Belvoir, VA: Defense Acquisition University.

Deininger, R. (1960). Human factors studies of the design and use of push-button telephone keysets. *Bell System Technical Journal, vol. 39*, pp. 995-1012.

Dekker, S. (2004). *Ten questions about human error*. Burlington, VT: CRC.

Dekker, S. (2006). *The field guide to understanding human error*. Burlington, VT: Ashgate.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (2004). *Human-computer interaction* (3rd ed.). Essex: Pearson Education Limited.

Dixon, S. R., & Wickens, C. D. (2006). Automation reliability in unmanned aerial vehicle control: A reliance-compliance model of automation dependence in high workload. *Human Factors, 48* (3), pp. 474-486.

DoD Standard: Flying Qualities of Piloted Aircraft, (1990).

DoD. (1998). *MIL-HDBK-759C: Human engineering design guidelines*. Washington, D.C.: U.S. Department of Defense.

DoD. (1999). *MIL-HDBK-1908B: Definitions of human factors terms*. Washington, D.C.: U.S. Department of Defense.

DoD. (2003a). *DoD directive 5000.01: The defense acquisiton system*. Washington, D.C.: US Department of Defense.

DoD. (2003b). *MIL-STD-1472F(1): Human engineering*. Washington D.C.: U.S. Department of Defense.

DoD. (2006). In OUSD (AT&L) (Ed.), *Risk management guide for DoD acquisition* (6th ed.). Washington, DC: Department of Defense.

DoD. (2008). *DoD instruction 5000.02: Operation of the defense acquisition system*. Washington, D.C.: US Department of Defense.

DoDAF Working Group. (2009a). *DoD architecture framework ver. 2.0, volume I: Definitions*. Washington, D.C.: Department of Defense.

DoDAF Working Group. (2009b). *DoD architecture framework ver. 2.0, volume II: Product descriptions*. Washington, D.C.: Department of Defense.

Dray, S. (1995, The importance of designing usable systems. *Interactions, vol. 2*, pp. 17-20.

Eamonn, K. S. (2004). Focus of attention and pilot error. *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications; ETRA '04,*

FAA. (2000). *System safety handbook.* Washington, DC: Federal Aviation Administration.

Federal Aviation Administration Order 7610.4, (2002).

Federal Aviation Regulations Part 91.113 and Part 91.11(a), (2005).

FAA. (2009). Aeronautical information manual (AIM). (Ch 6, Sec 6) Federal Aviation Administration.

Finkelstein, L. (2003). Widely, strongly, and weakly defined measurement. *Measurement, vol. 34*, pp. 39-48.

Fiorino, F. (2006). Fighting human error. *Aviation Week & Space Technology, 165*(22), pp. 47-49.

Fitts, P. M. (Ed.). (1951). *Human engineering for an effective air navigation and traffic control system*. Washington, DC: National Academy Press.

Flight Safety Technologies. (2009). *UNICORN.* Retrieved 15 Sep, 2009, from http://www.flysafetech.com/advancing_technologies/unicorn.asp.

Folds, D. J. (1995). Using operator role theory to guide function allocation in system development. *Proceedings of the Human Factors and Ergonomics Society, vol. 2,* pp. 1155-1159.

Francis, G. (1999). *A software tool for the design of multifunction displays* No. USAARL Report No. 99-20). Ft. Rucker, AL: US Army Air Research Laboratory.

Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (1994). *Feedback control of dynamic systems* (3rd ed.). New York: Addison-Wesley Publishing Company.

GAO. (2005). *GAO-05-301: Assessments of selected major weapon programs*. Washington, D.C.: U.S. Government Accountability Office.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York: W.H. Freeman.

Graeber, D., & Snow, M. (2008). In Hardman N. S. (Ed.), *Interview with boeing phantomworks engineers on HSI requirements*. Seattle, WA.

Hardman, N. S. (2006). Approaches for autonomous vehicles in civil airspace: Giving sight to the blind. (Master's Thesis, University of Tennessee).

Hardman, N. S., & Colombi, J. M. (2009). *A mapping from the DoD human error taxonomy codes to the domains of human systems integration* (Technical Report No. AFIT/EN/TR-09-04). Wright-Patterson AFB, OH: Air Force Institute of Technology.

Hardman, N. S., Colombi, J. M., Jacques, D. R., Hill, R. R., & Miller, J. E. (2009a). Application of a seeded hybrid genetic algorithm for user interface design. *Transactions of IEEE International Conference on Systems, Man, and Cybernetics,* San Antonio, TX.

Hardman, N. S., Colombi, J. M., Jacques, D. R., Hill, R. R., & Miller, J. E. (2009b). The challenges of human consideration in the systems engineering technical processes. *7th Annual Conference on Systems Engineering Research,* Loughborough University, UK.

Hardman, N. S., Colombi, J. M., Jacques, D. R., Hill, R. R., & Miller, J. E. (2009c). Persons in the processes: Human systems integration in early system development. *19th Annual International Symposium of INCOSE,* Singapore.

Hardman, N. S., Colombi, J. M., Jacques, D. R., Hill, R. R., & Miller, J. E. (Accepted for publication). An empirical function allocation method and its application to traffic collision avoidance. *International Journal of Applied Aviation Studies.*

Hardman, N. S., Colombi, J. M., Jacques, D. R., Hill, R. R., & Miller, J. E. (Submitted for publication). User interface design by a markov chain and hybrid seeded genetic algorithm approach. *IEEE Transactions on Systems, Man and Cybernetics--Part A: Systems and Humans.*

Hardman, N. S., Colombi, J., Jacques, D., & Hill, R. (2008a). Improved user interface design through graph-theoretic modeling. *IEEE Conference on Distributed Human-Machine Systems*, Athens.

Hardman, N. S., Colombi, J., Jacques, D., & Hill, R. (2008b). What systems engineers need to know about human-computer interaction. *INCOSE Insight, 11*(2), pp. 19-23.

Hardman, N. S., Colombi, J., Jacques, D., & Miller, J. (2008). Human systems integration within the DoD architecture. *The Industrial Engineering Research Conference,* Vancouver, BC.

Hardman, N. S., Jacques, D. R., Colombi, J. M., Hill, R. R., & Miller, J. E. (Submitted for publication). Requirements elicitation through legacy mishap analysis. *AIAA Journal of Aerospace Computing, Information, and Communication.*

Harris, D., & Muir, H. C. (Eds.). (2005). *Contemporary issues in human factors and aviation safety*. Burlington, VT: Aldershot.

Herron, E. (1978). Two types of system control with an interactive display device. *Society of Information Display Digest.*

HFIDTC. (2006). *HFIDTC home.* Retrieved May/30, 2007, from http://www.hfidtc.com.

Hoffman, R., & Elm, W. (2006). HCC implications for the procurement process. *IEEE Intelligent Systems, 21*(1), pp. 1541-1672.

Horrey, W., & Wickens, C. (2004). Driving and side task performance: The effects of display clutter, separation, and modality. *Human Factors, 46*(4), pp. 611-624.

ICAO Standard Annex 2: Rules of the Air, (1962).

ICAO Standard Annex 11: Air Traffic Services, Chapter 2; Appendix 4, (2001).

IEEE. (2008). *Systems and software engineering - system life cycle processes* No. IEEE STD 15288-2008)Institute for Electrical and Electronics Engineers.

Iglin, S. (2007). *Graph theory toolbox posted in matlab central.* Retrieved Aug 3, 2007, from http://www.mathworks.com/ matlabcentral/.

INCOSE. (2007). In Haskins C. (Ed.), *Systems engineering handbook* (ed. 3.1) International Council on Systems Engineering.

ISO. (2005). *Quality management systems--fundamentals and vocabulary* (9000:2005 ed.) International Organization for Standardization.

ISO. (2008). *A system and software engineering standard* No. ISO 15288:2008) International Organization for Standardization.

ISO/IEC. (2007). *Systems engineering - application and management of the systems engineering process.* ISO and IEEE.

Jaggers, T. (2005). *USAF systems engineering initiatives* (Briefing to NDIA 8th Annual Systems Engineering Conference. San Diego, CA: Deputy Assistant Secretary for Science, Technology, and Engineering.

JCS. (2007). *Joint capabilities integration development system* (CJCSI 3170.01F ed.). Washington, D.C.: Office of the Chairman of the Joint Chiefs of Staff.

Jean, G. V. (2009, July). Unmanned systems could fly majority of air force missions. *National Defense, 2009*, p. 34.

Johnson, C. (1999). Why human error modeling has failed to help systems development. *Interacting with Computers, 11*(5), pp. 517-524.

Kieras, D. (2004). Model-based evaluation. In A. Sears, & J. Jacko (Eds.), *Human-computer interaction handbook.* Taylor & Francis Group.

Kocay, W., & Kreher, D. (2000). *Graphs, algorithms, and optimization.* CRC Press.

Li, W. (2006). Pilot error and its relationship with higher organizational levels: HFACS analysis of 523 accidents. *Aviation, Space, and Environmental Medicine, 77*(10), pp. 1056-1061.

Lindlaw, S. (2008, 26 August, 2008). Drones' big enemy is human mistakes. *Arizona Daily Star,* p. 1.

Liu, B., Francis, G., & Salvendy, G. (2002). Applying models of visual search to menu design. *International Journal of Human Computer Studies, 56*(3), pp. 307-330.

Lockheed Martin. (2006) F-35 cockpit (proposed). *Code One.*

Maier, M. W. (1999). *Architecting principles for system-of-systems* No. CH1-460). New York, NY: John Wiley & Sons, Inc.

Malone, T. B., & Carson, F. (2003). HSI top down requirements analysis. *Naval Engineers Journal, 115*, pp. 37-48.

Martin, J. N. (1998). *Overview of the EIA 632 standard-"processes for engineering a system"* Briefing. San Diego, CA: EIA 632 Working Group.

Matsui, S., & Yamada, S. (2008). Optimizing hierarchical menus by genetic algorithm and simulated annealing. *GECCO '08: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation ; ACM.*

Mayhew, D. J. (1999). *The usability engineering lifecycle: A practitioner's handbook for user interface design*. San Francisco, CA: Morgan Kaufmann Publishers.

Michalewicz, Z. & Fogel, D. (1998). *How to solve it: Modern heuristics.* Springer.

Miller, C. A., & Parasuraman, R. (2007). Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control. *Human Factors*, *49*(1), pp. 57-75.

Mitchel, C. (1999). Model-based design of human interaction with complex systems In A. P. Sage, & W. B. Rouse (Eds.), *Handbook of systems engineering and management* (pp. 748-10). Somerset, NJ: John Wiley and Sons, Inc.

Mueller, M. (2009). In Hardman N. S. (Ed.), *Interview with personnel at the air force center for systems engineering*. Wright-Patterson AFB, OH.

Murray, J., & Reising, J. (1980). Tailoring software logic to the needs of the pilot: A software designer's nightmare? *IEEE Proceedings of the National Aerospace and Electronics Conference, ,* pp. 1052-1056.

Musselman, B. (2009). In Hardman N. S. (Ed.), *Interview with personnel of AFSC life sciences division*. Kirtland AFB, NM.

Narkevicius, J. M., Winters, J., & Hardman, N. S. (2008). Talking the talk: Cross-discipline terminology challenges. *INCOSE Insight, 11*(2), pp. 25-27.

NASA. (2007). *NASA/SP-2007-6105: Systems engineering handbook* (1st ed.). Washington, D.C.: National Aeronautics and Space Administration.

NDIA. (2003). *Top five systems engineering issues in defense industry* (Task Group Report. Arlington, VA: National Defense Industrial Association, Systems Engineering Division.

Newman, R. L., & Anderson, M. W. (1994). HUD flight testing: Lessons learned. *Southeast Section SETP Symposium,* Stone Mt., Georgia.

Norman, D. A. (2007). *The design of future things*. Jackson, TN: Basic Books.

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *Systems, Man and Cybernetics, Part A, IEEE Transactions*, *30*(3), pp. 286-297.

Payne, J., & Eppstein, M. (2005). A hybrid genetic algorithm with pattern search for finding heavy atoms in protein crystals. *Genetic and Evolutionary Computation Conference*, pp. 169-178.

Pew, R. W., & Mavor, A. S. (Eds.). (2007). *Human-system integration in the system development process*. Washington, D.C.: National Research Council.

Phillips, C. A. (2000). *Human factors engineering*. New York: Wiley & Sons, Inc.

Physiological Training Office. (2001). *USAF physiological training handbook*. Edwards AFB, CA: Medical Group Publication.

Puterman, M. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. New York: Wiley & Sons, Inc.

Reason, J. (1990). *Human error*. New York: Cambridge University Press.

Reason, J. (1997). *Managing the risks of organizational accidents*. Aldershot: Ashgate Press.

Reising, J. M., Calhoun, G. L., & Curry, D. G. (2009). In Hardman N. S. (Ed.), *Interview with personnel of AFRL advanced cockpit design team*. Wright-Patterson AFB, OH.

Reising, J. M., & Curry, D. G. (1987). A comparison of voice and multifunction controls: Logic design is the key. *Ergonomics*, *30*(7), p. 1063.

Rolfsen, B. (2009, 18 May). Pilot error cited in C-17 crash. *Air Force Times*, p. 25.

Rouse, W. B., & Rasmussen, J. (1981). Human detection and diagnosis of system failures. *NATO Symposiumon Human Detection and Diagnosis of System Failures.*

Sage, A., & Rouse, W. B. (Eds.). (1999). *Handbook of systems engineering and management*. New York: John Wiley & Sons, Inc.

Salvendy, G. (Ed.). (2006). *Handbook of human factors and ergonomics* (3rd ed.). New Jersey: John Wiley & Sons, Inc.

Sandia National Laboratories. (2004). *Revolutionary radar is five times smaller than existing technology.* Retrieved 15 Sep, 2009, from http://www.sandia.gov/news-center/news-releases/2004/def-nonprolif-sec/minisar.html.

Saunders, T. (2005). *System-of-systems engineering for air force capability development* (Technical Report No. SAB-TR-05-04)USAF Scientific Advisory Board.

Scerbo, M., W. (1996). Theoretical perspective on adaptive automation. In R. Parasuraman, & M. Mouloua (Eds.), *Automation and human performance: Theory and applications* (pp. 37-63). New Jersey: Erlbaum Assoc., Inc.

Schvaneveldt, R. W., Beringer, D. B., & Lamonica, J. A. (2001). Priority and organization of information accessed by pilots in various phases of flight. *11*(3), 253-280.

Seow, S. C. (2005). Information theoretic models of HCI: A comparison of the hick-hyman law and fitts' law. *Human-Computer Interaction, 20*(3), pp. 315-352.

Shappell, S. A., Detwiler, C., Holcomb, K., Hackworth, C., & Wiegmann, D. A. (2007). Human error and commercial aviation accidents: An analysis using the human factors analysis and classification system. *Human Factors, 49*(2), pp. 227-242.

Shappell, S. A., & Wiegmann, D. A. (2003). *A human error analysis of general aviation controlled flight into terrain accidents occurring between 1990-1998* (Technical No. ADA417230). Oklahoma City, OK: Federal Aviation Administration.

Sheridan, T. B. (2000). Function allocation: Algorithm, alchemy or apostasy? *International Journal of Human-Computer Studies, 52*(2), pp. 203-216.

Shneiderman, B., & Plaisant, C. (2005). *Designing the user interface* (4th ed.). Boston, MA: Addison-Wesley.

Sirevaag, E., Kramer, A., Wickens, C., Reisweber, M., Strayer, D., & Grenell, J. (1993). Assessment of pilot performance and mental workload in rotary wing aircraft. *Ergonomics, 36*(9), pp. 1121-1140.

Smith, W. J. (2000). *Modern optical engineering* (3rd ed.). Boston: McGraw-Hill Professional.

Society of Automotive Engineers (SAE). Aerospace Recommended Practice ARP5580: *Recommended Failure Modes and Effects Analysis (FMEA) Practices for Non-Automobile Applications*, June 2000.

Stanton, N. A., Harris, D., Salmon, P. M., Demagalski, J. M., Marshall, A., Young, M. S., et al. (2006). Predicting design induced pilot error using HET (human error template)

- A new formal human error identification method for flight decks. *Aeronautical Journal, 110*(1104), pp. 107-115.

Stanton, N. A., Salmon, P. M., Walker, G. H., Baber, C., & Jenkins, D. P. (2005). *Human factors methods: A practical guide for engineering and design*. Burlington, VT: Ashgate Publishing Lmtd.

Stimson, G. W. (1998). *Introduction to airborne radar* (2nd ed., pp. 136). New York: SciTech Publishing.

Tan, K. K., Qing-Guo, W., & Chieh, H. C. (1999). *Advances in PID control*. London, UK: Springer-Verlag.

Thompson, W. T., Tvaryanas, A. P., & Constable, S. H. (2005). *U.S. military unmanned aerial vehicle mishaps: Assessment of the role of human factors using HFACS*. Brooks City-Base, TX: 311th Performance Enhancement Directorate, United States Air Force.

Tvaryanas, A. P. (2006a). *Human factors considerations in migration of unmanned aircraft system (UAS) operator control* No. AD-A444925; HSW-PE-BR-TR-2006-0002; XC-311TH HSW). Hanover, Maryland: NASA Center for Aerospace Information.

Tvaryanas, A. P. (2006b). Human systems integration in remotely piloted aircraft operations. *77*(12), pp. 1278-1282.

U.S. Air Force Studies Board. (2008). *Pre-milestone A and early-phase systems engineering: A retrospective review and benefits for future air force systems acquisition*. Washington, D.C.: The National Academies Press.

UAVM. (2009). *Collision avoidance: "detect sense and avoid" or "see and avoid"*. Retrieved 15 Sep, 2009, from http://www.uavm.com/uavregulatory/ collisionavoidance.html.

USAF. (1998). *AFI 91-202: Air force safety investigation requirements* (Air Force Instruction No. AFI 91-202). Washington, D.C.: Office of the Secretary of the Air Force.

USAF. (2006). *AFI 91-204: Safety investigations and reports* (Air Force Instruction No. AFI 91-204). Washington, D.C.: Office of the Secretary of the Air Force.

USAF. (2007). *AFI 63-1201: Life cycle systems engineering* (Air Force Instruction). Washington, D.C.: US Air Force.

USAF HSI Office. (2008). *Briefing to the secretary of the air force*. Arlington, VA: U.S. Air Force HSI Office.

USAF TPS. (2000a). Flight control systems. *Flying qualities phase handbook* (). Edwards AFB, CA: USAF Test Pilot School.

USAF TPS. (2000b). Forces of flight. *Performance phase handbook* (pp. 4-1). Edwards AFB, CA: USAF Test Pilot School.

USAF TPS. (2000c). Sensors. *Systems phase handbook* (pp. 4-113). Edwards AFB, CA: USAF Test Pilot School.

USAF TPS. (2000d). Sensors. *Systems phase handbook* (pp. 4-41 & 4-189). Edwards AFB, CA: USAF Test Pilot School.

USAF TPS. (2000e). Sensors. *Systems phase handbook* (pp. 4-20). Edwards AFB, CA: USAF Test Pilot School.

von Bertalanffy, L. (1968). *General systems theory* (2nd ed.). New York: George Braziller.

Warwick, G. (2004, 27 Apr). Remote chance. *Flight International, 2004*, p. 1.

West, D. B. (2001). *Introduction to graph theory* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall, Inc.

White House Commission on Aviation Safety and Security. (1997). *Final report to the president* (Final No. per Executive Order 13015). Washington, DC: White House.

Wickens, C. (2002). Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science, 3*(2), pp. 159-177.

Wickens, C. D., Lee, J., Liu, Y., & Gordon-Becker, S. (2004). *An introduction to human factors engineering* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Wiegmann, D. A., & Shappell, S. A. (2001). Human error analysis of commercial aviation accidents: Application of the human factors analysis and classification system (HFACS). *Aviation Space And Environmental Medicine, 72*(11), pp. 1006-1016.

Wiegmann, D. A., & Shappell, S. A. (2003a). *A human error analysis of commercial aviation accidents using HFACS* (Technical No. DOT/FAA/AM-01/3). Washington, DC: Office of Aviation Medicine, Federal Aviation Administration.

Wiegmann, D. A., & Shappell, S. A. (2003b). *A human error approach to aviation accident analysis*. Burlington, VT: Ashgate Publishers.

Williams, K. W. (2006). *Human factors implications of unmanned aircraft accidents: Flight-control problems* (Technical No. ADA460892). Oklahoma City, OH: Federal Aviation Administration.

Wirthlin, J. R. (2009). In Hardman N. S. (Ed.), *Interview with personnel air force materiel command*. Wright-Patterson AFB, OH.

Wolfe, R. C. (2004). *NASA ERAST non-cooperative DSA flight test* (Final No. 2004-6). Edwards AFB, CA: National Aeronautics and Space Administration.

REPORT DOCUMENTATION PAGE						*Form Approved* *OMB No. 074-0188*
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**						
1. REPORT DATE *(MM-YYYY)* 11-2009			**2. REPORT TYPE** Doctoral Dissertation			**3. DATES COVERED** *(From – To)* Aug 2006 – Dec 2009
4. TITLE AND SUBTITLE AN EMPIRICAL METHODOLOGY FOR ENGINEERING HUMAN SYSTEMS INTEGRATION						**5a. CONTRACT NUMBER**
						5b. GRANT NUMBER
						5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Hardman, Nicholas S., Major, USAF						**5d. PROJECT NUMBER**
						5e. TASK NUMBER
						5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865						**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/DS/ENV/09-D01
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) 711th Human Performance Wing, Human Effectiveness Division						**10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RH**
						11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT The systems engineering technical processes are not sufficiently supported by methods and tools that quantitatively integrate human considerations into early system design. Because of this, engineers must often rely on qualitative judgments or delay critical decisions until late in the system lifecycle. Studies reveal that this is likely to result in cost, schedule, and performance consequences. This dissertation presents a methodology to improve the application of systems engineering technical processes for design. This methodology is mathematically rigorous, is grounded in relevant theory, and applies extant human subjects data to critical systems development challenges. The methodology is expressed in four methods that support early systems engineering activities: a requirements elicitation method, a function allocation method, an input device design method, and a display layout design method. These form a coherent approach to early system development. Each method is separately discussed and demonstrated using a prototypical system development program. In total, this original and significant work has a broad range of systems engineer applicability to improve the engineering of human systems integration.						
15. SUBJECT TERMS Human systems integration (HSI), human-computer interaction (HCI), user interfaces, systems engineering						
16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** John M. Colombi, AFIT/ENV	
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3355, ext 3347 (john.colombi@afit.edu)	
U	U	U	UU	247		

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18